clyd
ControL Your Devices

User Manual

Version 7.0.0

Table of Contents

Chapter 1. Introduction

This manual includes all Clyd features, including some that are not essential for a first-time user, or which may only be useful in very specific situations. For ease of reading, the chapters concerned are marked as "advanced".

To make it easier for you to learn and read, this manual includes brief presentations of features marked with a symbol

Some of the features presented impose prerequisites that do not correspond to the situations of all Clyd users. These are indicated throughout the document by a symbol

Some Clyd features are explained in this manual but are not always visible because they are deprecated. These are indicated by the symbol

For all your needs, from assistance in getting started with Clyd to feedback on this manual, please contact Telelogos support:

- Tel : +33 (0)2 41 22 70 18
- support@telelogos.com

Chapter 2. I want to install Clyd on my server (advanced)

2.1. Installation prerequisites

Android 6 to Android 15. Android Go not supported. ***Note: Android versions 10 to 14 are only supported with Android Enterprise. Android 6 is only supported in Device Admin mode.***

Using the Chrome browser for Windows to access the administration console. Simultaneous use on multiple tabs is supported.
Microsoft Windows Server 2016 / 2019 / 2022
Microsoft SQL Server 2012 / 2014 / 2016/ 2017 / 2019 / 2022 / 2025
Microsoft Internet Information Server
Installing Microsoft .Net Framework 6
Microsoft .Net Core 3.1 (installed by setup)

Installing IIS 10 (Windows 2016)
If you have not already done so, you must install IIS 10. To see if this component has been installed, you can simply look under the Server Manager to see if "IIS" appears in the list of installed roles.
To install IIS 10:
- Use the "Server Manager."
- Click "Add Roles and Features" in the "Manage" menu.
- Click through the first screen "Before you begin."
- Click through the second screen "Installation Type."
- Click through the third screen "Server Selection."
- Check the "Web Server (IIS)" role.
- In the pop-up window that opens, click the "Add features" button.
- Click "Next."
- Click "Next" once more.
- Click through the introductory screen to the "Web Server" role.
- Select "Web Server / Application Development/ASP.NET 6" from the list of services.
- Select "WebSocket Protocol" from the list of services.
- In the pop-up window that opens, click the "Add features" button.
- Click "Next" and then "Install."

MediaContact Server (Version 6.15.0 or higher)
CLYD does not support Multi-server mode or Persistent connection mode.
MediaContact Console (Version 6.15.0 or higher)
MediaContact Web Services (Version 6.15.0 or higher). You must have already installed Windows Hosting Bundle .NET 6 (search Hosting Bundle on the page).

> *i* If you are using Clyd on a web page and you do not need to install Clyd on a server, you should pass this chapter and start your reading at the next chapter.

2.2. Proxy

CLYD does not support PAC proxy files. In order to configure a proxy, you must manually configure it in IIS:
- If the proxy is configured directly on the CLYD website, then this configuration will be lost during the next CLYD update.
- If the proxy is configured on the parent node of the CLYD website and the CLYD website inherits it, then this configuration will not be lost during the next CLYD upgrade.

2.3. Connections in HTTPS

HTTPS connections to the CLYD web server, the clyd-iot web service (for Zebra printer support) and MediaContactWebServices require the purchase and installation of a TLS certificate on the web server.

2.4. Compatibility between MediaContact and CLYD

Version table

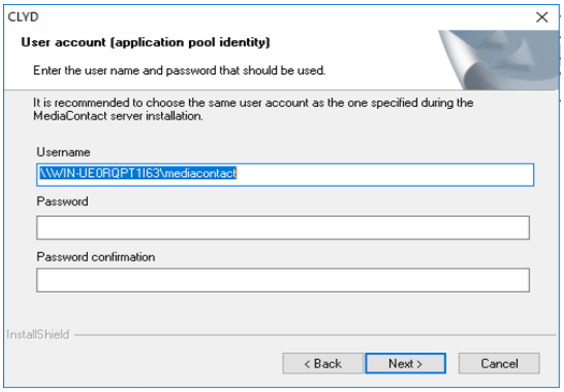
CLYD version	Release date	MediaContact version (minimum)
3.1.2	November 16, 2017	6.3.4
3.1.3	April 19, 2018	6.4.0
4.0.0	July 6, 2018	6.4.1
4.1.0	September 30, 2018	6.4.5
4.2.0	December 15, 2018	6.4.6
5.0.0	March 31, 2019	6.5.3
5.1.1	March 9, 2020	6.6.5
5.2.0	August 13, 2020	6.7.1
5.2.1	September 30, 2020	6.7.2
5.2.3	December 3, 2020	6.7.2
5.2.4	January 20, 2021	6.7.2
5.3.0	January 31, 2021	6.7.5
5.3.1	April 21, 2021	6.7.6
5.4.0	July 9, 2021	6.8.1
5.5.0	February 18, 2022	6.9.0
5.5.1	May 5, 2022	6.9.1
5.6.0	June 27, 2022	6.9.2
5.6.1	July 22, 2022	6.9.2
6.0.0	December 22, 2022	6.10.0
6.0.1	March 15, 2023	6.10.2
6.1.0	June 16, 2023	6.12.1
6.2.0	December, 2023	6.13.0
6.2.1	February, 2024	6.13.1
6.2.2	March, 2024	6.13.2
6.2.3	April, 2024	6.13.2
6.2.5	June, 2024	6.13.2
6.3.0	September, 2024	6.14.0
7.0.0	June, 2025	6.15.0

2.5. Installing the CLYD V7.0.0 Server

Installing the "Web Console" component

Run … \Console web\setup.exe

Application pool identity: Use the "MediaContact Server" service account



In Internet Information Services (IIS):

Create a virtual directory "> *Default Website* > *CLYD*"
Create the associated application pool "*CLYDPool*"

Test the connection to the CLYD console:
http://localhost/CLYD
User name : admin
Password : admin

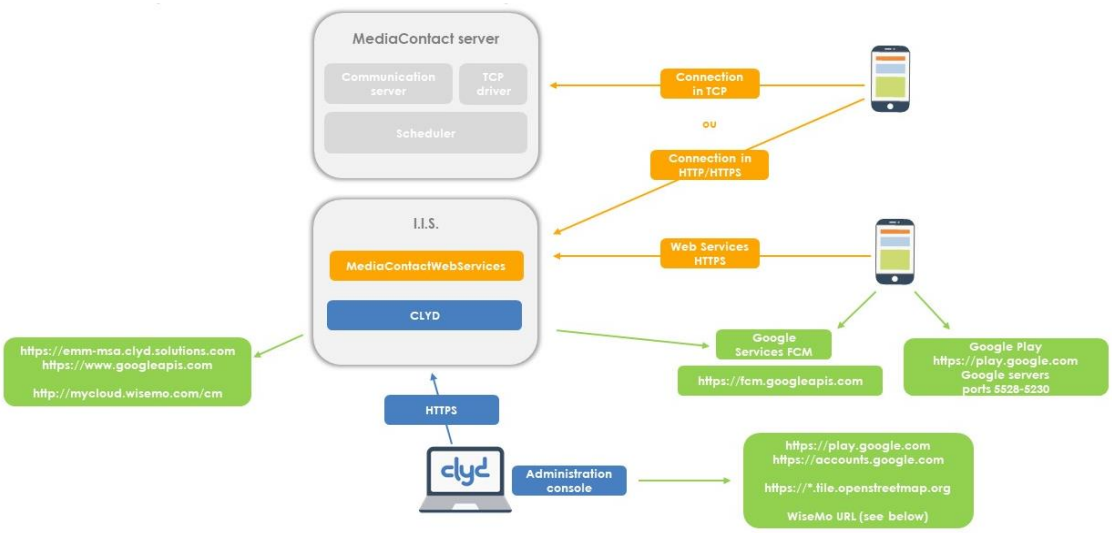
During a new installation (MediaContact + CLYD), the following items are configured by default:

The MediaContact server is partitioned.
A company "My company" is created
A connection schedule (default: every 4 hours between 06:00 and 22:00 with a 3 hours call distribution) is created.
MediaContact internal security is enabled (administrator account: admin/password: admin).
Authorize automatic device creation: Activated
Authorize automatic device reassignment: Activated
Open the activity window at each communication: Deactivated
Show communication errors in a message window: Deactivated

Specific case of CLYD installation on a running MediaContact server:
The MediaContact settings are unchanged
Warning: Only the MediaContact "Global Administrator" account is allowed to connect to CLYD.
It will therefore be necessary to manually configure the MediaContact server (see above, case of a new installation).

2.6. Architecture diagram

2.6.1. CLYD with ANDROID ENTERPRISE



FIREWALL SERVER/DEVICES/CONSOLE

CLYD console				
Protocol	**Source**	**Destination**	**Port**	**URL**
HTTP/HTTPS	CLYD console	CLYD Web Server	80/443	-
HTTP/HTTPS	CLYD console	MediaContactWebServices	80/443	-
CLYDMediaContact communications				

Protocol	Source	Destination	Port	URL
TCP	Devices	MediaContact Server	1300	-
	Devices	Communication server	1310	-
Or				
HTTP/HTTPS	Devices	MediaContactWebServices	80/443	-

Communications with Google Android Enterprise servers

For more details, see the Android Enterprise Network Requirements page.

Protocol	Source	Destination	Port	URL
HTTPS	CLYD console	Google servers	443	https://play.google.com
HTTPS	CLYD console	Google servers	443	https://accounts.google.com
HTTPS	CLYD Web Server	Telelogos server	443	https://emm-msa.clyd-solutions.com
HTTPS	CLYD Web Server	Google servers	443	https://www.googleapis.com
HTTPS	Devices	Google servers	443	https://play.google.com

Device communications with Google servers

Protocol	Source	Destination	Port	URL
HTTPS	Devices	Google servers	443, 5228-5230	List on the Google website

(optional) Real-time feedback from devices via Web Services (kiosk status, inventories)

Protocol	Source	Destination	Port	URL
HTTP/HTTPS	Devices	MediaContactWebServices	80/443	-

(optional) Contacting devices in real time via FCM Google Services (using Google ID)

Protocol	Source	Destination	Port	URL
HTTPS	CLYD Web Server	Google servers	443	https://fcm.googleapis.com

(optional) Display the map (geolocation)

Protocol	Source	Destination	Port	URL
HTTPS	CLYD console	OpenStreetMap servers	443	https://*.tile.openstreetmap.org

(optional) Enable remote control of devices via WiseMo myCloud

Go to the page Firewall settings for minimal myCloud access

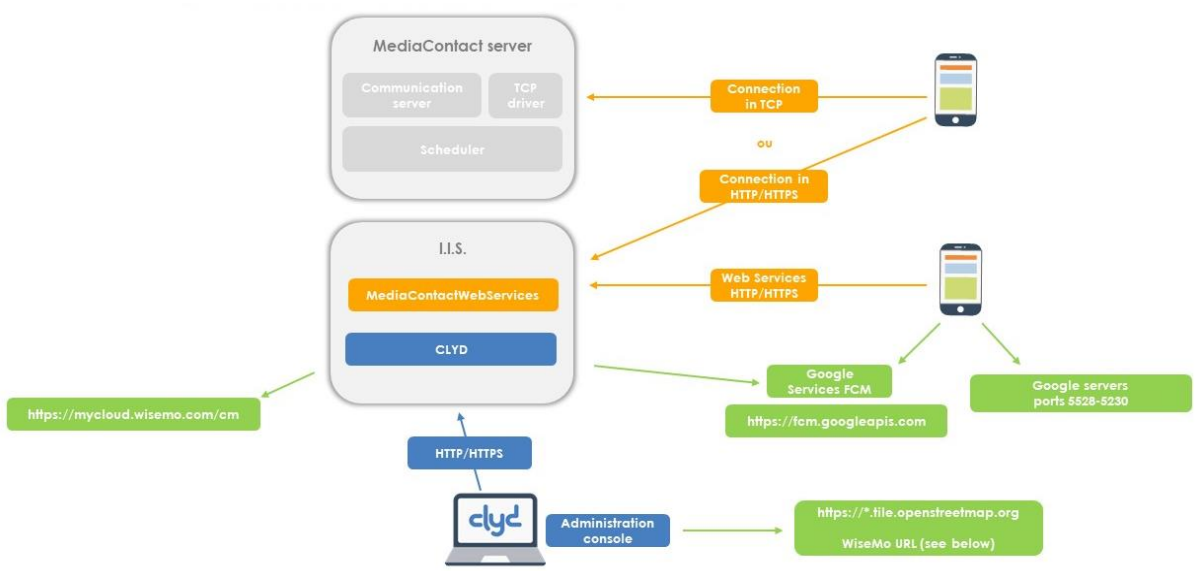
(optional) Contact Samsung servers

Go to the page How do I allow Knox services to contact Samsung servers?

(optional) Using the REST API to manage the CLYD server

Protocol	Source	Destination	Port	URL
HTTPS	Application	CLYD Web Server	443	-

CLYD without ANDROID ENTERPRISE



FIREWALL SERVER/DEVICES/CONSOLE

All the previous rules except the "communications with Google Android Enterprise servers" ones.

2.6.3. Protect your server access

By default, the Clyd console is accessible on the Internet via the URL https://nom_de_domaine/CLYD.

As detailed in the diagram in 2.6.1 and 2.6.2, access from terminals to the server is via the URL https://nom_de_domaine/MediacontactWebServices.

It is possible to block external access to the Clyd console on IIS. To do this, you can block https://nom_de_domaine/CLYD/*. **Please note** that company and terminal enrollments, as well as public APIs, will no longer be usable via the Internet unless the following URLs are authorized:

- If you need to enroll a company, you'll also need to open the Google callback URL: https://nom_de_domaine/CLYD/api/Companies/completeEmmSignup
- If terminal enrollments are carried out from the "Installation package" console page and via the Internet, the following URL must be authorized: https://nom_de_domaine/CLYD/xtr/CustomPackages/*
- To use the public APIs, open https://nom_de_domaine/CLYD/public/api/*

Chapter 3. I want to register / modify my Android company

Android Enterprise lets you control devices and declare yourself device owner to Google. This is what establishes the link between your devices and your company for Google.

Clyd allows you to enrol and manage "fully managed device", either controling devices with a kiosk or allowing access to settings by default and settings security restrictions and applications (semi-open mode).

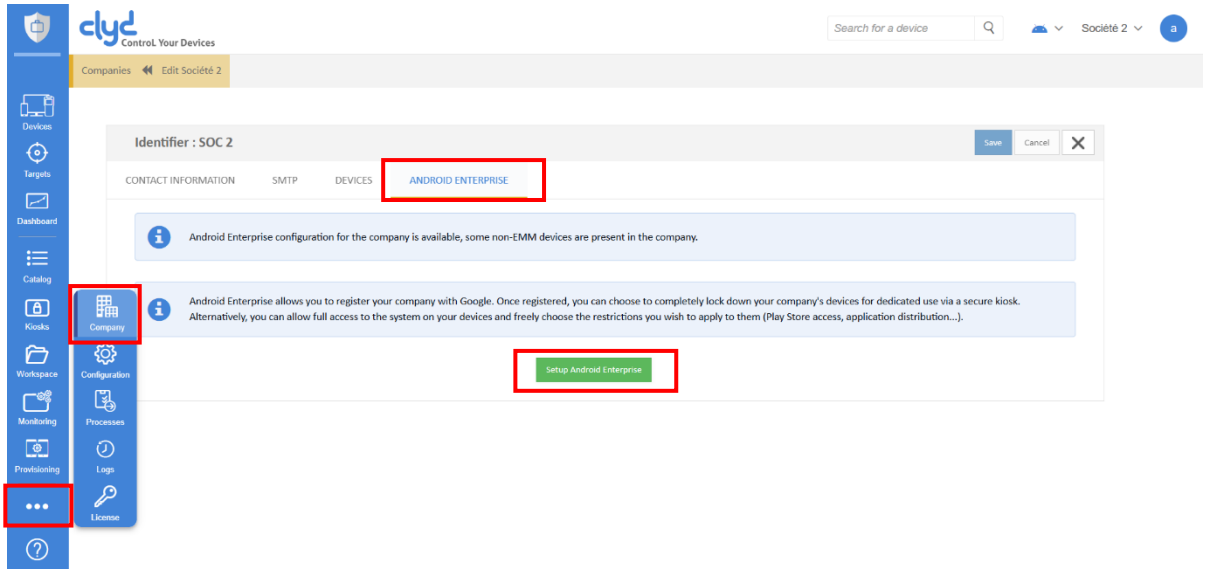
In ClydTo manage an Android company, you must use devices from Android 6 or plus and a version of the Clyd client (CLYDMediaContact) V6.4.x minimum. **To use Android Enterprise, you need devices from Android 7 or plus (recommended).**

If you are using Clyd on your own server (advanced), in addition to the above requirements you must:

- Have MediaContact Server V6.4.x minimum, MediaContact WebServices V6.4.x minimum and CLYD Console Web V4.0 minimum.
- Use HTTPS (requires a certificate for the IIS server): connection to the CLYD console in HTTPS mode and connection of devices in HTTPS.
- Use Chrome browser for Windows. Use on multiple tabs simultaneously is supported

3.1. I want to link my company to my Clyd account

When you create your first Clyd account, you must select the Android Enterprise registration mode you want in the "Android Enterprise" tab of the "Company" menu.



Once you have made your selection, click on "Setup Android Enterprise".

You will then be redirected to the Google Play console, which allows you to manage your enterprise with Google (which you can still access via https://play.google.com/work/adminsettings). You'll need a Google account with your own login and password (Clyd has no knowledge of these).

At Google, you will need to complete the registration process: declare your company name, accept the Google Play enterprise contract, then complete the registration process.

When you return to Clyd, you will see in the "Android Enterprise" tab that your company is now registered, and you'll find your username, the previously registered administrator email, and the chosen registration mode.

3.2. I want to modify my Android registered company

On the Google Play console (https://play.google.com/work/adminsettings), you can:

- Check your Android Enterprise company details (ID and name). This information cannot be modified.
- Verify and modify declared contacts.
- Modify the emails designated as owners and administrators of your company and add new ones.

Clyd only knows the username, company name and email address of the owner registered when your Android Enterprise (Google) entity was created.

Only you can declare other owners/administrators of your entity to Google. You must always ensure the management (addition, deletion) of the declared owner(s) of your entity with Google, while maintaining the confidentiality and security of these identifiers.

Additional information on company registration: (advanced)

Registering your CLYD Company as an Android Enterprise requires a connection to the GOOGLE server using the private keys of the software publisher (the publisher is certified by GOOGLE as an EMM partner).

When you request to register a CLYD company as an ANDROID ENTERPRISE from the CLYD console (Company Menu – ANDROID ENTERPRISE tab), the CLYD server connects to the TELELOGOS server which stores the private keys. Then it is the TELELOGOS server that directly requests GOOGLE to register the CLYD company, and the GOOGLE server sends back validation of the company registration to the CLYD server (sending a company identifier visible in the ANDROID ENTERPRISE tab).

CLYD OUTGOING

URL for the TELELOGOS server for the private keys:
Only used during the enrollment of the company to Android Enterprise
https://emm-msa.clyd-solutions.com

URL for the GOOGLE server for the Android Enterprise Web Services:
Used for all Android Enterprise function calls (Enrolling the company, redirecting to the Google Play Store, etc.)
https://www.googleapis.com

3.3. I want to unregister a CLYD company (advanced)

Caution: unregistering a company has several immediate impacts, particularly on your applications, application management and devices. If in doubt, contact your support team before unregistering.

When a CLYD company is enrolled in Android Enterprise, a button in the Android Enterprise tab allows you to unenroll it (menu “Company”).

The information linked to the Android Enterprise account and the devices managed are conserved for 30 days by GOOGLE after unenrollment.

3.4. I want to define the SMTP used to send Clyd emails

In the SMTP tab (“Company” menu), you can set the SMTP used to send geofencing alerts. Without this setting, no e-mail alerts are sent.

3.5. I want to view or modify my Clyd company information

Contact details:

You can fill in your company's contact details, as well as define contact details for a correspondent and an administrator.

Devices:
Total number of devices associated with the company, including assigned and decommissioned devices.

Chapter 4. I want to enrol a device

Enrolment establishes a link between the Clyd server and your device. There are several steps to enrol a device:

- **Generating an installation package**
  The installation package contains all the information needed to install the CLYDMediaContact client and to register the device with the CLYD server.
  The package consists of the CLYDMediaContact client, configuration information and automatic device registration. It may also contain a manufacturer add-on (which provides access to certain advanced functionalities), your in-house applications and your files for saving in the device.

- **Installing the package**
  The package must be installed to the device (See the different methods below).

- **Registering the device with the CLYD server.**
  The device is automatically registered (you can view your device details in the CLYD console).

4.1. I want to choose an enrolment mode

i The following presentation of all enrolment modes is not essential for your first enrolment, so you can skip this section and go straight to the next section.

Android Enterprise Devices
Your company must be registered with Google Android Enterprise (see Chapter 1 page 12).

QR-CODE (Android 7 to Android 15)
- Reset device to factory settings
- Tap 6 times on the screen
- Scan the QR-CODE to install the DPC and the CLYDMediaContact client

ADB (Android 7 to Android 15)
- Enable developer options on device and enable USB debugging
- Connect the device to be registered over USB on the host PC
- Download the registration package in zip format on the host PC and unzip the archive
- Launch the execution of the install_adb.bat file

Token EMM (Android 7 to Android 15)
- Reset device to factory settings
- Enter Gmail address: afw#clyd
- CLYD DPC (download and automatic installation)
- From the CLYD DPC: scan the QR-CODE to install the CLYDMediaContact client

Zero Touch (Android 8 to Android 15)

Warning: you can only use the Zero Touch mode if your device and device supplier are authorized for Zero Touch by Google. SAMSUNG devices cannot use ZERO TOUCH mode.

On the Google portal
- Connect to the Google Zero Touch portal (https://enterprise.google.com/android/zero-touch/customers/)
- Create a new configuration
  - EMM DPC: select CLYD DPC
  - Extra DPC: copy and paste the field “DPC extras used in the Zero Touch portal”

- Associate this configuration with your devices

On the device
- Reset device to factory settings
- Configure a network (cellular data or WiFi)
- Download and automatically install the CLYD DPC and CLYDMediaContact client

KNOX MOBILE ENROLLMENT/KME (Android 8 to Android 15)

Warning: You can only use KME mode if your SAMSUNG devices are eligible to use this registration mode.
See: https://support.samsungknox.com/hc/en-us/categories/115001758008

Non-Android Enterprise Devices (Android 6 to Android 9)
DEVICE ADMIN mode

Once the package has been generated, you can:

Download the package (Direct download)
The package is downloaded to your workstation. You must copy this package to the device (via USB) in order to execute it.

Scan the QR-CODE (Link via QR-CODE)
Go to the QR-CODE reading application on your device and flash the QR-CODE to download the package directly to the device and execute it.

4.2. Generating an installation package

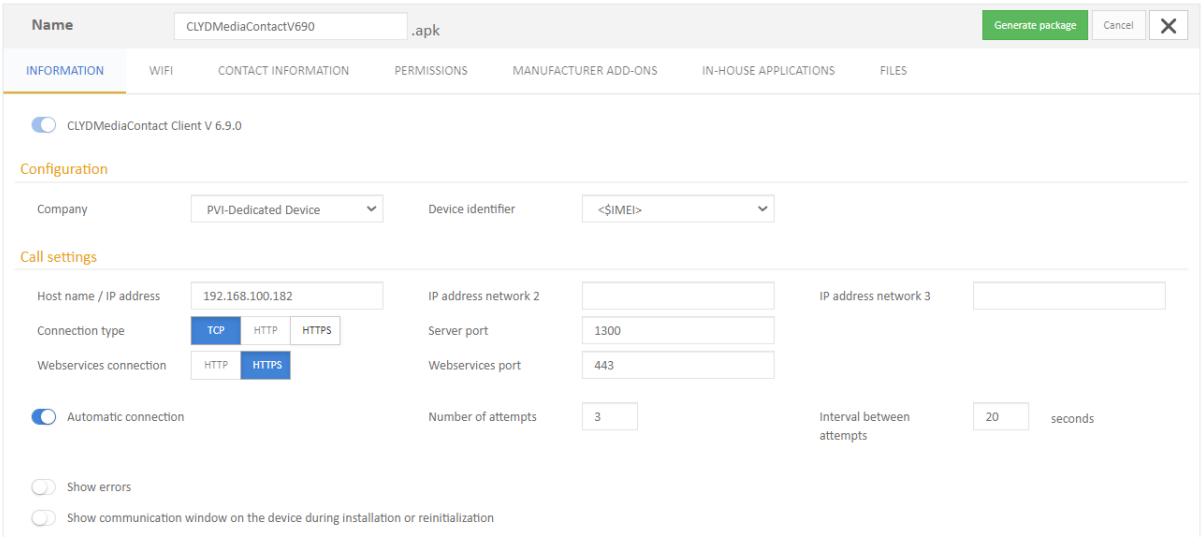
The interface can be accessed from the "Configuration" menu, then "Installation packages". It shows all packages already created. You can, at any time:
- Modify an existing package by clicking on its name.
- View the QR-code of your package by clicking on the eye.

To create an installation package, you must select the type of package you want to generate by clicking on the "+" sign. Then you can set parameters in your package and "generate" the package.

i For your first smartphone or tablet enrolment, we recommend that you use QR-Code enrolment and then refer to the Getting Started Tutorial (available on https://www.telelogos.com/clyd-telechargements/) as it is not mandatory to modify the additional parameters detailed below.

When you create a package, you will then be taken to the interface shown below.

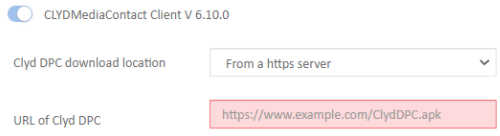
❖ Tab: Information



- **Client CLYDMediaContact** Vx.x : Latest client version available on the CLYD server.
  For a new package, the version of the CLYDMediaContact client indicated at the top of the tab is the latest version available on the CLYD server.

  For an existing package, the version of the CLYDMediaContact client indicated at the top of the tab is the version currently available in the package. If a more recent version is available on the CLYD server, a warning message appears. The package must be regenerated to include this new version.

- **Clyd DPC download location (QR code only)**



  If the devices that will use this package do not have access to the Google Play Store for downloading the Clyd DPC, it is possible to specify another download location:
  From the Clyd server: in this case the device will download the Clyd DPC directly from the Clyd server;
  From a https server: in this case the device will download the Clyd DPC at the URL you entered in the "Clyd DPC URL" field.

- **Device identifier** : Unique data to identify the device within the company. You can keep the default information (the identifier of your device is the IMEI No.) or choose one of these identifiers :

  $Auto : Sequence number assigned by the CLYD server
  $IpAddress : Device IP address
  $HostName : Device name
  $LoginName : Name of the Gmail address
  $MacAddress : MAC address for the active interface
  $IMEI : International Mobile Equipment Identity (recommended)
  $UUID : Universally Unique Identifier (RFC 4122)
  $Serial : Hardware serial number, if available.
  $AndroidID : 64 bit identifier generated by Android

- **Call settings (**advanced, see 4.2.1after**)**
- **Installation settings on the device**

  Enable all system applications (By default: Enable)
  When using the Android Enterprise mode, all system applications on the device (Camera, barcode reader, etc.) are disabled by default. Use this setting to enable/disable these system applications after installing the CLYDMediaContact client.

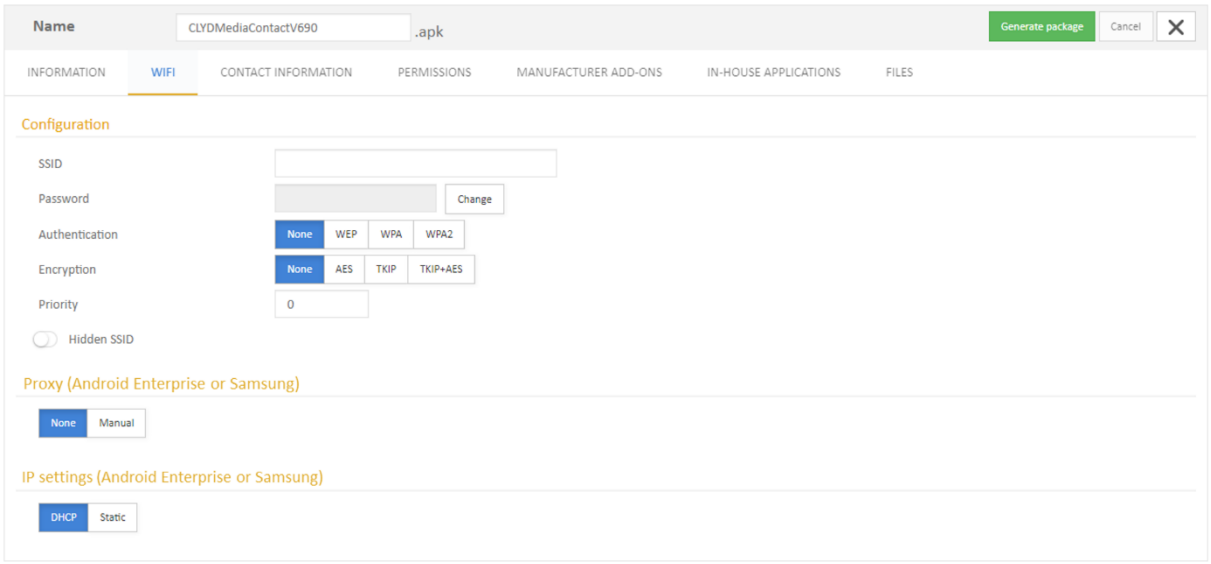
  Ignore external SD card detection (By default: Enable)

This setting blocks the use of the external SD card.
On some devices (GAIA, Honeywell), this setting will prevent the internal card as being falsely recognized as an external SD card. This can generate file synchronization problems.

Save the installation package contents on the device
This setting allows you to copy your files to a specific folder (to be specified in the interface). It also allows you to store the APK files of your in-house applications on the device.

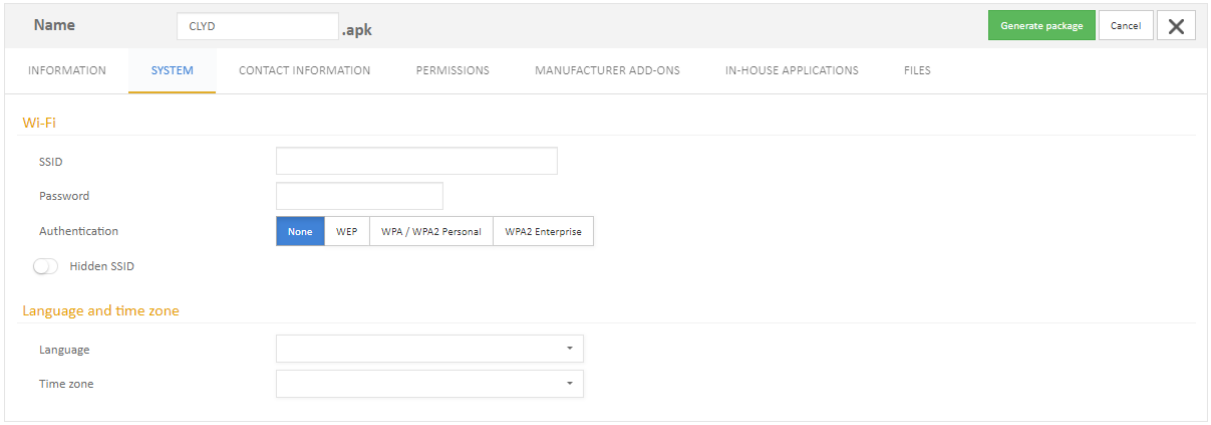
Use mobile data for device registration (QR-Code, Zero Touch/Android 10 to Android 13)
Use this setting to force the use of mobile data during the registration phase. This is useful when there is no available WiFi network at the time of registration. Its use by the Android system depends on the manufacturer's implementation of this setting, as well as the mobile data operator.

❖ **Tab: Wi-Fi (ADB, EMM Token, Zero Touch, Samsung KME, Device Admin)**



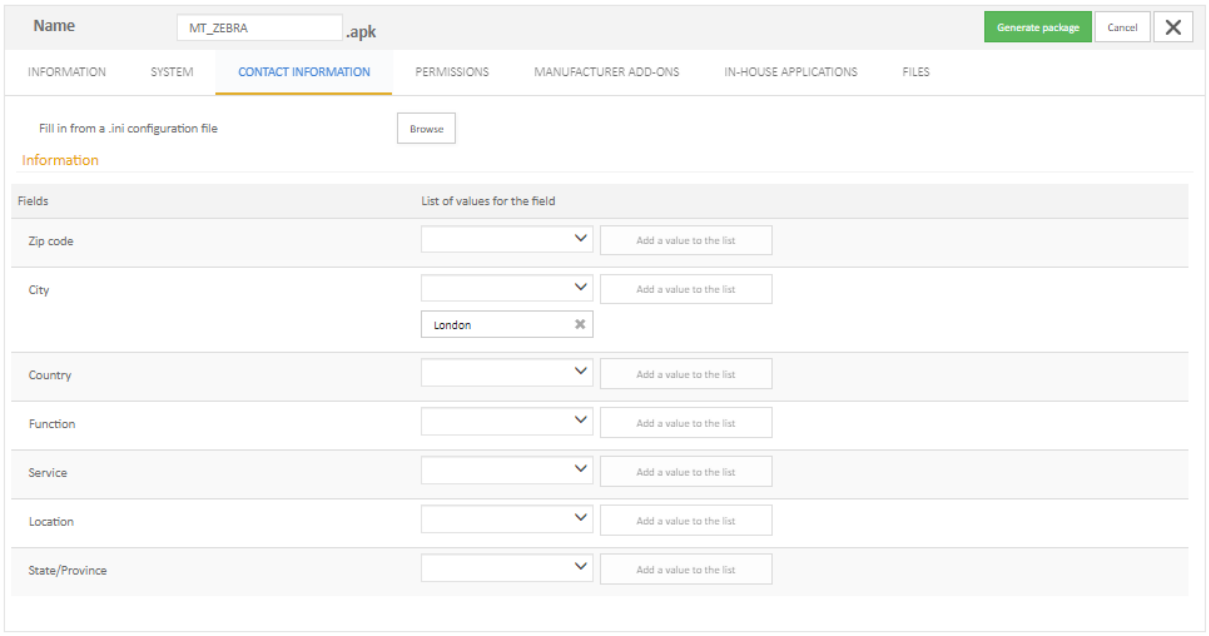
You can configure a WiFi network for your device, this wifi will be used for the for connexion between the device and the server.
You can configure proxy and ip settings (for Samsung or Android Enterprise devices).

❖ **Tab: System (QR-Code)**



You can configure a WiFi network, the language and the time zone for your device. This information is required in order to download the DPC and the CLYDMediaContact client.

❖ Tab: Contact information



This tab improves the feature previously offered by the "enrollment.ini" file before version 5.5, making it simpler and more intuitive. Older packages that used the "enrollment.ini" file automatically benefit from this new tab.

This functionality allows you to populate the address fields of the device when installing the package so that the device can download the correct kiosk/WorkSpace or EMM Profile during registration or when resetting the CLYDMediaContact Client.

The available fields are zip code, city, country, function, service, location and state/province.

Using an "enrollment.ini" file:
If you have a correctly formatted "enrollment.ini" file, you can import it to automatically populate this tab. The enrollment.ini file can contain one or more fields among the coordinate fields:

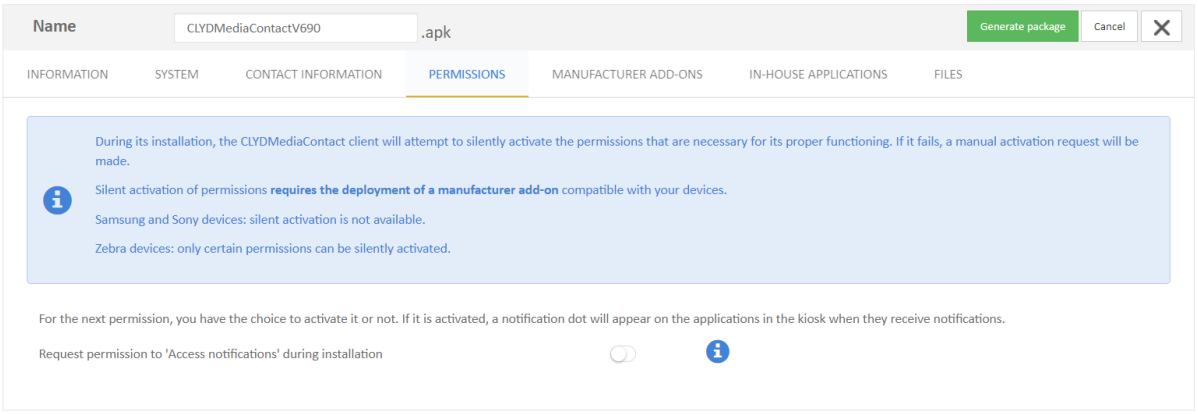
- Zip code (field PostCode)
- Country (field Country)
- City (field City)
- State/Province (field State)
- Function (field UserFunction)
- Location (field Localisation)
- Service (field Service)

Example:
UserFunction="FUNCTION 1";"FUNCTION 2";"FUNCTION 3"

Enrolling a device or resetting the CLYDMediaContact client:
If you add several values to a field in the "Contact information" tab of the package, then the technician in charge of deploying devices will have to choose one of these values when initializing the CLYDMediaContact client. This value will appear in the device's file on the server.

If you add a single value to a field in the "Contact Information" tab of the package, then the technician in charge of deploying the devices will not have to enter anything. This value will automatically appear in the device's file on the server.

❖ Tab: Permissions



During its installation, the CLYDMediaContact client will attempt to silently activate the four permissions listed below. If it fails, a manual activation request will be made. These permissions are necessary for CLYD's proper use.

Silent activation of permissions requires the deployment of a manufacturer add-on compatible with your devices.

Samsung and Sony devices: silent activation is not available.

Zebra devices: only "Ignore battery optimizations", "Access usage data" and "Overlap with other applications" permissions can be silently activated.

Change system settings (Android Enterprise and Device Admin)
Grant this permission to authorize managing the device's standby mode (sleep, wake, sleep time), screen rotation and immediate blocking of the device when a password policy is applied.

Ignore battery optimizations (Android Enterprise and Device Admin)
Grant this permission to allow the device to run processes in the background when the device is in deep sleep mode (e.g. daily call during the night).

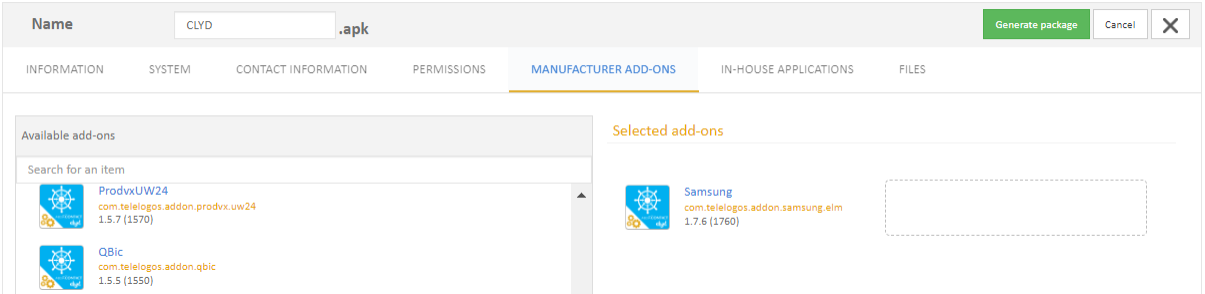
Access usage data (Android Enterprise and Device Admin)
Grant this permission to authorize the kiosk to block/allow specific applications/activities.

Overlap with other applications (Device Admin)
Grant this permission to authorize the kiosk to block the system tray.

For the next permission, you have the choice to activate it or not. If it is activated, a notification dot will appear on the applications in the kiosk when they receive notifications.

Access notifications (Android Enterprise and Device Admin)
Grant this permission to authorize the display of notification dots in the kiosk.

❖ Tab: Manufacturer add-ons



Select the manufacturer add-on for your device. A search box is used to filter available add-ons based on the contents of the add-on's name and package name.

❖ **Tab: In-House applications**

Select your In-House applications from the list of available applications or download them directly. Your application(s) will be installed before CLYDMediaContact. They are installed in alphabetical order.

❖ **Tab: Files**

Select the files to add to your package. All file types can be added to the package. "Device installation settings" allows you to define the folder where file copies are stored on the device.

4.2.1. I want to modify installation parameters (advanced)

i In the « informations » tab of installation packages, it is possible to set some installation advanced calling parameters.

Host name/IP address : IP address or DNS name of your CLYD server
IP address network 2 :
IP address network 3 :

If the CLYD server is configured for Internet and Intranet access, multiple addresses can be specified.

Connection type : Communication protocol between the CLYD server and the devices.

The devices can connect to the CLYD server either in HTTP (default port: 80), Https (default port 443) or TCP (default port 1300 and 1310) mode.

Web Services Connection : Used to provide information feedback in real time (e.g. kiosk status).

The virtual directory/Default Website/MediaContactWebServices depends on the port used by Default Website (default port: 80). See configuration of the Internet Information Server.

Automatic connection : Automatically connect to the CLYD server after client installation.

Initialization connection: The CLYDMediaContact client must connect to the CLYD server after installation to register the device with its company, download its configuration and upload its inventories. You can define a number of connection attempts and a time period between two attempts.

Install the root certificate of the CLYD server's certification authority (CA)

Available for QR-Code packages in Android Enterprise companies, this option is used to deploy the root certificate of the CA which generated the CLYD server certificate. It will be installed on the devices before the package is downloaded. This is required if the CA is not known to the devices (e.g. company CA).
Warning: The QR-Code generated with this option is composed of a high density matrix of 177 by 177 black square modules. This means the devices to be registered must have a high quality camera to correctly scan this QR-Code. Displaying the QR-Code on a larger screen or projecting it onto a larger surface may help reading the QR-Code on some low quality cameras.

Authorize remote access : Enable mobile data before the initialization connection.

If the initialization connection requires Mobile data to be enabled, the CLYDMediaContact client can be requested to do so by providing an access point name (APN).

Show errors : Show communication errors during the initialization communication.

Display the Communication window : Display an activity monitor during the initialization communication.

4.2.2. I want to enrol my devices without access to the Clyd server (advanced)

> This feature is for partners who prepare devices with the CLYDMediaContact client, but cannot connect to the CLYD server during the preparation process and wish to secure access to the device.

To prepare a package containing a kiosk, use a "Connection to the CLYD server" widget. When the device is delivered, it is locked by this kiosk. On site, the technician can then launch a connection to the CLYD server in order to register the device in their company account, then download the client's kiosk with their in-house applications.

Kiosk:
Create a kiosk containing a "Connection to the CLYD server" widget. Retrieve the kiosk.xml file from the CLYD server and insert it as a file into the package.
(C:\mediacontact\mcs\mediacontact\MYDProfiles\Company Name\Kiosk Number\kiosk.xml)

At the end of the CLYDMediaContact client installation, if the device cannot connect to the CLYD server, this kiosk will be launched.
When the device reboots, this kiosk will secure your device and allow you to connect to the CLYD server (Widget). During this initial connection, the kiosk configured in CLYD for this device will replace the kiosk included in the package. If no kiosk has been configured in CLYD, the kiosk in the package is deleted.

4.2.3. I want to distribute certificates when enrolling my device (advanced)

> This feature allows you to distribute and install certificates on the device during registration. It is also possible to deploy certificates via a process using the prov_cert command (see 13.3).

Two types of certificates can be installed:

- Root certificates from certification authorities: the file must contain the certificate from the certification authority in X509 Base64-encoded (PEM) or X509 binary-encoded (DER) format, and its extension must be .cer or .crt.
- PKCS#12 archives including a user certificate and the private key associated with the certificate. It is usually password protected. The file must be in PKCS#12 format and its extension must be .p12 or .pfx. **Warning:** Up to Android 9, the device must have an active lock mode (PIN, lock pattern, password, etc.) in order to install PKCS#12 user certificates. Once the user certificates are installed, disabling the device lock mode will automatically remove the installed user certificates. This constraint therefore prevents the deployment of PKCS#12 user certificates via the installation package on devices up to Android 9. For Android 10 to Android 13, this constraint no longer exists.

In "Device Admin" mode, on Samsung devices, certificate provisioning is only available on Android 9.

Here are the steps to follow to deploy one or more certificates on a device:

1) Create a file named "certificate.xml" containing information related to the certificates to be installed. The %RootSystemFolder%\MCS\MediaContact Android\certificate.xml file contains the syntax for the file and an explanation of the various settings. You can use it as an example for writing your own:

    - The *path* field contains the path to the certificate.
    - The *alias* field contains the name that will be associated with the user certificate for a PKCS#12 archive during installation (unused field for a root certificate).

- The *password* field is required when using a PKCS#12 archive to extract the private key if it is password protected (recommended) (field not used for a root certificate).

2) Add the "certificate.xml" file to the CLYD file catalog, as well as the various certificates that you wish to install on the device, according to the formats and naming standards indicated previously.
3) Add these files to the package.
4) Deploy the package.

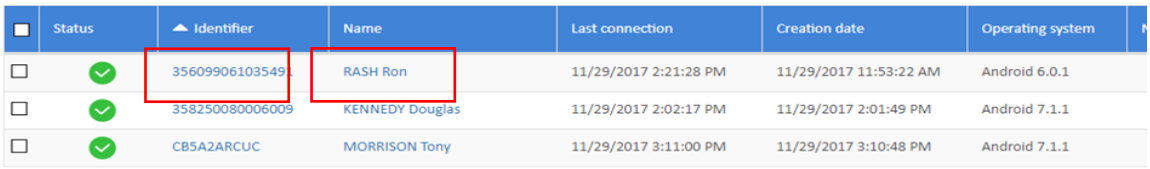
Sample of the certificate.xml file

```
<?xml version="1.0" encoding="UTF-8"?>
<certificate>
 <!--Install a CA certificate-->
 <entry>
  <path>/sdcard/Download/Root CA.crt</path>
 </entry>
 <!--Install a user certificate and private key from a PKCS#12 archive-->
 <entry>
  <path>/sdcard/Download/Android1.p12</path>
  <alias>Android1</alias>
  <password>password</password>
 </entry>
</certificate>
```

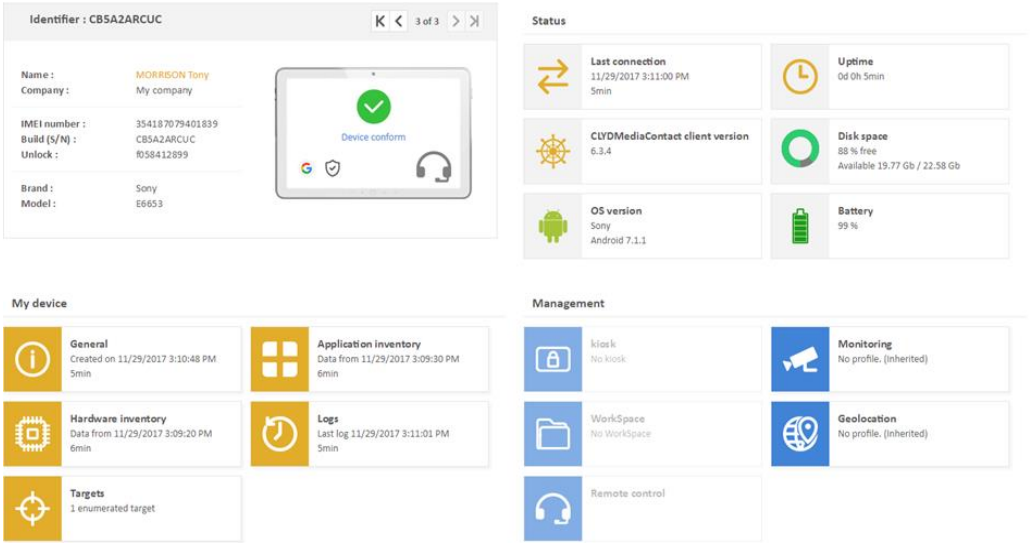
4.3. I want to confirm that my device is enrolled

Access to the device list (Menu – Devices)

Once the CLYDMediaContact package is installed, the device automatically connects to the CLYD server to register with its company. A Hardware and Applications inventory is fed back during this connection. The device list displays the device and allows you to view the device information sheet.



Access to the device details (by clicking on the identifier or the name), see Chapter 10 page 42.



4.4. I want to understand how device registration works in detail

For a device to access Android Enterprise features, a Device Policy Control (DPC) must be installed before installing the CLYDMediaContact client. The DPC is found in the Google Play Store. These actions are performed automatically by the package.
The CLYDMediaContact client package must be prepared on your CLYD server, via the "Configuration > Generate a package" option.

WARNING: When creating the package (Information tab)
Call setting – Connection type TCP or HTTPS
Call setting – WebServices connection HTTPS

Information to check, following the enrollment of your device:

Check on the device

On the information page on the device, check the following entries are present:
EMM device Id : Value1
EMM afw account : valeur2@android-for-work.gserviceaccount.com
EMM device owner (CLYD DPC is device owner)

Check on the console

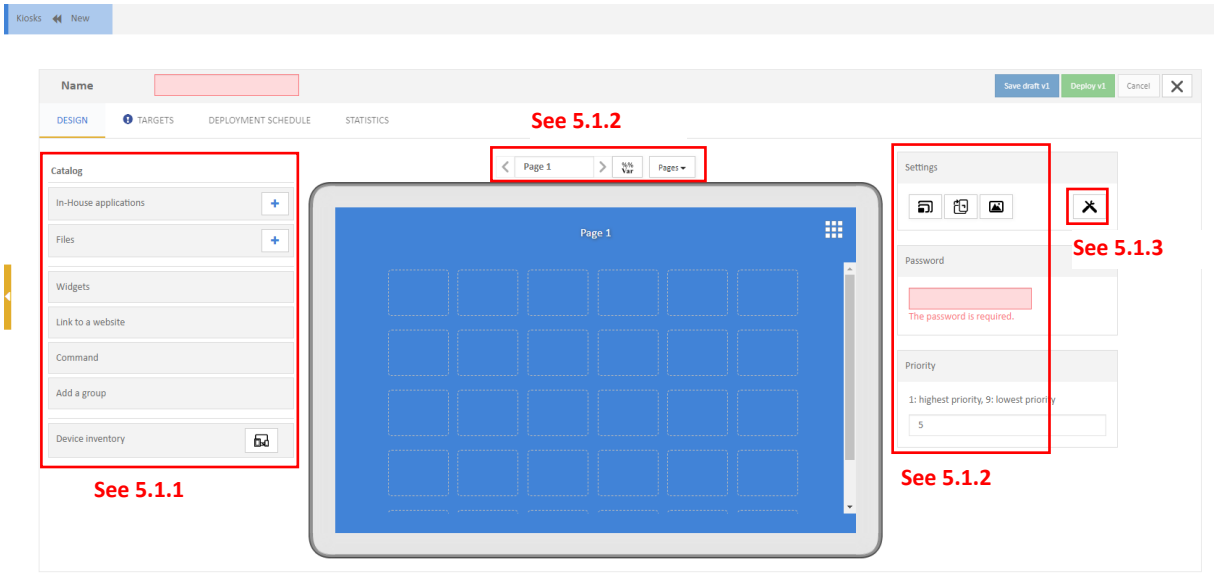
In the device information tab, check the following entries are present:
EMM User Id : Value3
EMM device Id : Value1

Chapter 5. I want to lock my devices and restrict its uses (kiosk mode)

Clyd can activate a kiosk mode on the device that limits access to a defined list of programs. Access to the Android operating system is no longer possible, so the device is secure by default.
The kiosk is displayed in full-screen mode when the device starts up and the programs are presented with their icon and label specified in the design phase.
To return to normal mode (complete access to the device's system), an administrator password must be entered when closing the kiosk (tap 3 times on the kiosk to access this menu).

If you want to managed devices with specific restrictions and free access to the system, you should set semi-open devices with a workspace if you want to deploy apps and a security profile (see the next chapters)

5.1. Designing a kiosk



5.1.1. Adding items to the kiosk

Add items: To add elements to the kiosk, simply drag and drop them or click on one of the elements. A group can be added to group several elements (up to 16).
Deleting items: Elements can be deleted from the kiosk at any time by placing the pointer over an element and clicking on the recycle garbage can, or by dragging and dropping another element onto the element to be deleted.
Move elements: The display order of elements defined in design is the order in which these elements will be displayed on the kiosk.

Catalog
You can select applications or files from your catalog. Applications added to the kiosk will be installed on the device after an installation request is sent to the Google servers (enterprise store applications) or after running the APK file (in-house applications). Files added to the kiosk will be copied to the devices in the "/sdcard/CLYDFiles" folder.
An application or a file can be downloaded in this menu; they will be loaded in the catalog and available directly to be saved to your kiosk.

Files added to the kiosk will be copied to device in the "/sdcard/CLYDFiles" folder.
You can download applications and files from this menu, which will then be loaded into the catalog and made available directly to your kiosk.

Special case of files already present on the devices: to display on the kiosk a shortcut to a file already present on the device, you must create a command containing the full access path to this file (e.g. "/sdcard/DCIM/ photo.jpg"). The use of the "*" wildcard character is possible (e.g. "/sdcard/DCIM/*.jpg"). In this case, a shortcut for each file present and corresponding to the criteria is created on the kiosk. When a new file meeting the criteria is added in the same folder, a new shortcut is automatically created on the kiosk.

Apart from "File" items, the label and icon of each item on the kiosk can be customized using the pencil icon overlapping the item. This customization will of course be reflected in the kiosk displayed on the devices.

Widget
Wi-Fi config: Connect to a Wi-Fi by entering the password.
Wi-Fi: Activate or deactivate
Bluetooth: Pair the device with another equipment
Cellular data: Activate or deactivate (needs a manufacturer signed addon)
Brightness: Adjust brightness
Telephone: Grant access to the telephone function
Message: Have a message containing variables.
Sound: Adjust the volume + -
Locate: Activate or deactivate
Auto-rotate: Activate or deactivate
Connection: Make the device connect to the Clyd server
Flashlight: activate or deactivate

Link to a website
You can grant access to a website by specifying a URL

Use this field to launch:
A MediaContact Station process: process:"process name"
A shortcut to an html page: file://sdcrad/file.html

Add a command (advanced – see 13.3 page 58)
Use to add commands and intents.

Add a group
Use to create a group where you can place up to 16 icons (applications, files)

Device inventory
You can select a device to view its Software Inventory, and grant access to one of the applications in the kiosk. The application must be installed on the device. It will not be installed by CLYD. If it is not present on the device, it will not be displayed on the device's kiosk.

Caution: If an application is added to the Clyd catalog, it will be deployed on all devices targeted by a kiosk that previously integrated it via Device Inventory. However, there is an option to prevent the application from appearing on devices that do not have the application in their inventory in this case: when editing the application, activate "Do not deploy the application on devices (when present in the same version in the In House catalog)".

5.1.2. Configuring the kiosk

Create pages

You can configure tabs to distribute your applications/files/widgets, etc. across multiple pages.
You can change the page name by specifying a label or using a variable.

Screen resolution

You can adapt the kiosk displayed in Clyd to the resolution of your devices, either by indicating a resolution or by selecting a device.

Screen orientation

Portrait/landscape mode for your kiosk design

Style settings

You can add wallpaper, select a background color and select a color for apps names and text (title and apps name).

Password

Set a password necessary for kiosk exit. By default, 3 clicks are required in the kiosk to access the kiosk exit menu on the device (can be set in advanced kiosk properties, see 5.1.3).

Priority

Setting priority levels on your kiosks allows CLYD to start the highest priority kiosk in the event that multiple kiosks are deployed on one device (in the case of non-disjoint targets, for instance).
The highest priority is 1, the lowest priority is 9 and the default priority is 5.
If multiple kiosks deployed on a device have the same priority level, the kiosk deployed last will be started on the device.
Examples:

- Three kiosks are deployed on a single device. Kiosk A has priority 6, kiosk B has priority 2 and kiosk C has priority 9 →Kiosk B is started because it has the highest priority.
- Three kiosks are deployed on a single device. Kiosk A has priority 6, kiosk B has priority 6 and kiosk C has priority 2 (Kiosk C is started because it has the highest priority.
- Three kiosks are deployed on a single device. Kiosk A has priority 2, kiosk B has priority 2 and kiosk C has priority 9 (the most recently deployed of kiosks A and B is started.

5.1.3. Advanced properties of the kiosk

Advanced properties can be set by clicking on the icon.

- **Access to the general kiosk settings.**
  You cand define the number of clicks necessary to exit the kiosk on the devices.
  You can also set some other options, for instance prohibit access to the system tray, block Safe Mode, or block device rotation.
  Caution, if rotation is blocked, the rotation defined from the Clyd interface will be applied to the terminal (if the kiosk design is defined in portrait, the terminal will have a portrait display and vice versa). By default, if this option is not enabled, the terminal display is independent of the kiosk design display from Clyd.
  You can also choose to position elements (applications, widgets, commands, etc.) in your kiosk manually or automatically.

- **Authorize or block certain windows.**
  By default, the kiosk does not allow any windows to occupy the foreground, except those belonging to the specified applications. Nevertheless, it may be necessary to enable certain windows to show in the foreground (e.g. a password or PIN entry window). Similarly, it may be necessary to forbid certain windows from showing in the foreground (e.g. a window opened by a third-party application, providing access to configuration of the device).

  To make the kiosk allow all activities of all applications, simply create an authorized application with the wildcard character * in both the "Application full path (package)" and "Window name (activity)" fields. In this case, you can still unauthorize certain applications/activities by adding them to the list of unauthorized applications.

  The advanced setting "Hide "Banned Apps/Windows" alert messages, when disabled, displays a message over the kiosk whenever an app/activity is blocked.

  Applications/activities blocked by the kiosk are systematically logged in a cyclic file by the CLYDMediaContact client, whether the advanced setting "Hide warning messages from prohibited Applications/Windows" is activated or not. This file is accessible on each device, after activating the "Configuration -> Extract configuration" menu of the CLYDMediaContact client, at the following location: /sdcard/MCExtract/tmp/kiosk_unauthorized_apps.txt.

- **CLYD Browser**

Using the built-in CLYD browser, which allows you to specify a particular behavior, for example, to lock the address bar, allow domains / prohibit URLs or clear cache.

When the "Clear cache" option is enabled, the cache will be emptied each time the browser is opened. If set to "single application" mode, it will be cleared on return to the home page when the "Return to home page on inactivity" option is enabled.

Notifications:
By default, you can view notifications on the kiosk icons. If you block the "system tray", you can also let users access the details of their notifications without having access to the "settings" in the system tray (Android 9 to Android 13).

- **Device blocking/warning**

Blocking in driving mode
You can block access to the device if you exceed a certain speed (from 5 to 99 km/h), access will be unlocked if you fall below this speed for a certain time (between 5 and 60 seconds).
Only calls are notified and the "pick up" and "hang up" buttons are allowed on the terminal when blocking is activated and a call is incoming.
Two types of locking can be chosen:
A speedometer is displayed. No other information is visible
A transparent screen allowing the device to be viewed (e.g. a GPS navigation application)
Bluetooth pairing works, Telephone calls can be received in hands-free mode.

Device alerts on connection
You can set up an alert that your users receive on their device if they have not connected to the server after a certain time. No blocking is performed on devices, and the alert is only displayed on your users kiosk, not when they are using a kiosk application. If the device connects to the server, the alert disappears in the next minute.

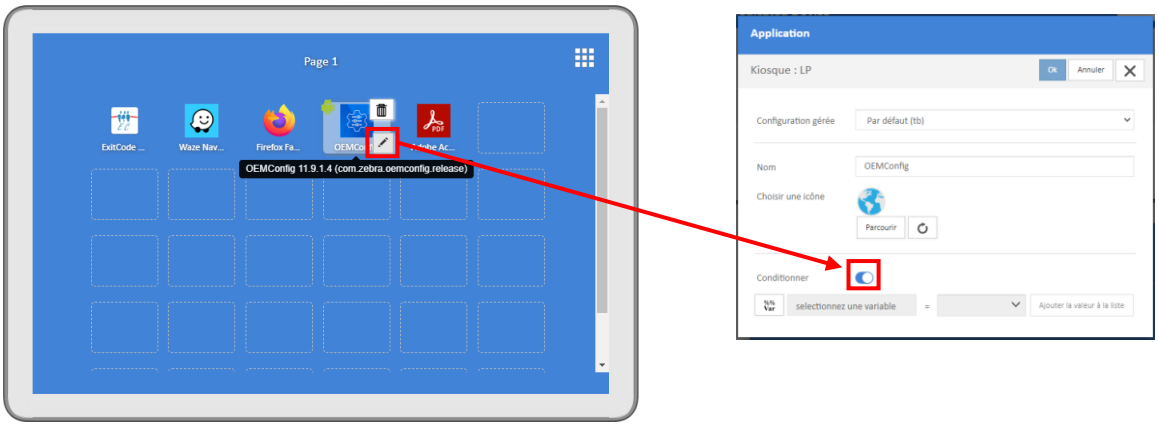
You can configure the message your users will see and the delay after which they receive this alert. You need to insert "%CONNECTION-DELAY%" in your text to display the delay set on Clyd to your user.

- **Authentication (advanced)**
You can define an In-House application from your catalog as the kiosk authentication application. In this case, it is automatically deployed on the relevant devices when the kiosk is deployed.
The authentication application automatically launches. It blocks access to the kiosk until the user successfully authenticates, in the following cases:
  - Authentication failed
  - Kiosk startup (manual or automatic);
  - Device wakeup, when the kiosk is activated.

In order to be used as an authentication application, an application must meet certain criteria to interface with the kiosk. These technical specifications will be provided by Telelogos upon request.

5.1.4. Condition the display of kiosk elements on terminals

It is possible to condition the display of all or some of the elements added to the kiosk. This ensures that only suitable applications are displayed on terminals receiving the same kiosk (and thus avoids the risks associated with multiple kiosks with only minor differences). Clyd also allows you to define different kiosks sent to different targets.



To condition the display of a kiosk element, position your cursor on the element, then edit it, and activate display conditioning. Choose a variable from among the terminal's coordinates and aliases, and one or more conditions from among the known values for this variable. The item in question will then only be displayed on terminals meeting this condition.

Conditions can be set for all elements of a kiosk or group. They cannot be set for a group.

5.2. TAB: Security

Security elements are now defined as a full-fledged provisioning profile from the "Provisioning" menu (see 12.2 page 51). This feature has been depreciated since v6.1 (June, 2023) and will be discontinued.

The "Security" tab no longer exists for new kiosks since Clyd 6.1.0 and will disappear for existing kiosks. This is why it is advisable to create a security provisioning profile using the elements defined in this tab and targeting the same endpoints. This profile creation can be done by clicking on the "Create a security profile" button. The creation of this security profile uses the security parameters, targets and distribution schedule of the current kiosk.

Please note that the security provisioning profile only applies to devices with Clyd DPC 6.1 and the ClydMediaContact 6.12 client at least. It is therefore important to update your fleet of devices and then distribute your security profile.

Once these operations have been carried out, it is advisable to remove the security parameters present in the kiosks by clicking on the "Delete" button because they will no longer be effective.

Note that in the event of a conflict between the security settings defined in the Kiosk security tab and the security profile in the "Provisionning" menu, the settings in the provisioning menu "security profile" always take precedence over those defined in the security tab.

5.3. Deploying a kiosk

If you want to save your kiosk without deploying it, you can click on "Save Vx draft." You can also duplicate your kiosk in the kiosk list.

You cannot deploy if you have not selected a target. When you add targets in the TARGETS tab, the "Deploy Vx" button will be available.

Immediate deployment.

This is the default deployment mode.
Click on the "Deploy Vx" button to deploy version x of your kiosk. If the devices have a Google-ID, the kiosk will be deployed immediately, otherwise it will be deployed the next time the device connects to the CLYD server (see 15.1).

Catalog applications (excluding inventory applications) will be copied and installed on the device when the kiosk is deployed.

Catalog files will be copied and installed on the device when the kiosk is deployed.

Scheduled deployment.

You can define a validity period in the Deployment Schedule tab. Google-ID is not used. The kiosk will be deployed to devices that connect during the configured validity period. The applications will be copied and installed when the kiosk is deployed. The files will be copied during the kiosk deployment.

Example: validity period, starting today, from 22:00 to 06:00 every day except Saturday and Sunday.

The devices connect to the CLYD server according to the connection schedule, therefore a connection schedule must be configured to ensure the devices connect during this period.

Stopping deployment.

Whether the deployment in progress is manual or scheduled, it can be stopped at any time by clicking on the "Stop Vx deployment" button.

If you are using managed configurations, deploying a kiosk has the effect of publishing the selected managed configurations for the applications in your kiosk to all devices in the targets associated with the kiosk.

5.4. File synchronization

The catalog files saved to the kiosk will be copied when the kiosk is deployed. You can update these files automatically. You just need to update the file in the catalog and select a synchronization mode in the deployment schedule.

Synchronizing files "at each communication"
All files saved to the kiosk will be synchronized each time the device connects to the CLYD server.

Synchronizing files "at the first communication in the validity period"
All files saved to the kiosk will be synchronized when the device connects during the validity period. If the device connects multiple times during the validity period, the files will only be synchronized once (the first time).

5.5. Updating an application

To update an application, you need to deploy a new kiosk with the new version of the application.
You can either modify your current kiosk (it will stop the deployment in progress) or duplicate it to keep the original version. Duplicating a kiosk allows you to test the new application on a restricted target.

In the catalog or directly in the kiosk design tab in the Catalog menu, download the new version and save it to the kiosk before deploying.

5.6. Deleting the kiosk on the device

A kiosk will be deleted from a device in the following situations:

Deleting a kiosk: You can delete one or more kiosks from the kiosk list. The kiosk will be removed from the device the next time it communicates with the CLYD server.

Deleting a target: You can delete the target containing your device in the kiosk. The kiosk will be deleted from the device at the next deployment (A delete request will be broadcast to the device).

Deleting the device in a target: If you delete your device from a target that is used in a kiosk, the kiosk will be deleted from the device at the next deployment (A delete request will be broadcast to the device).

Warning: deleting a device from a company or unassigning a device does not remove the kiosk from this device. You need to uninstall the CLYDMediaContact client from the device or reset the CLYDMediaContact client.

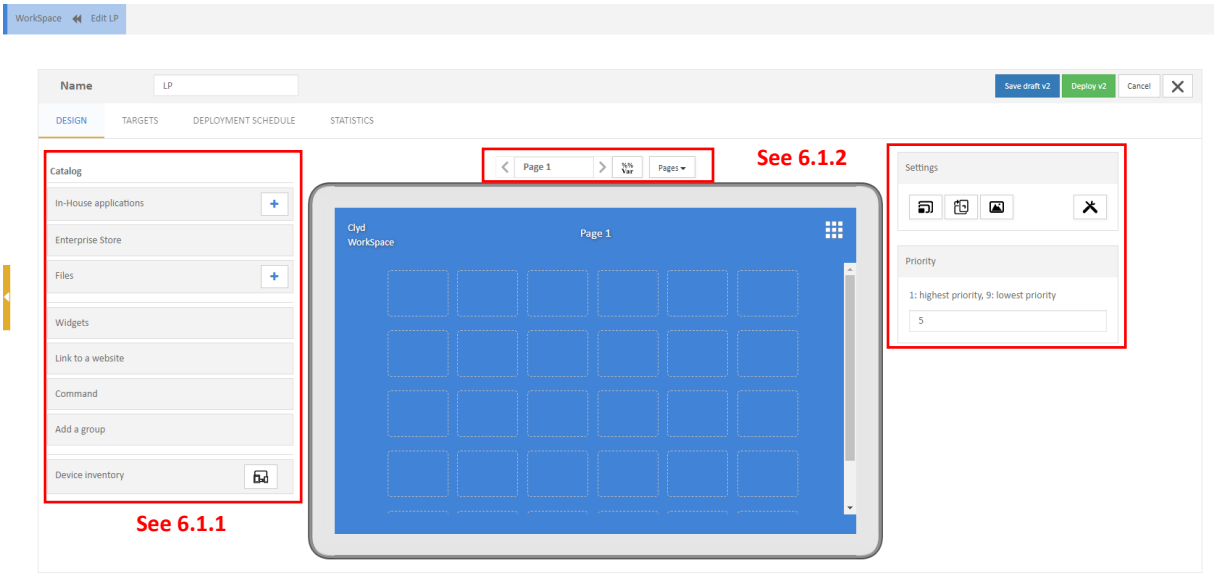
Chapter 6. I want to set up my background administrator area (WorkSpace)

CLYD can activate a WorkSpace mode on the device. The WorkSpace will be available (once synchronized) in the device applications. It provides the device user with access to all his/her applications and in-house files in a single application.

It can be deployed without a kiosk, but can also be combined with a kiosk, to set up a background administration area invisible to users but useful for operators and administrators. In this case, the elements of the WorkSpace are invisible without password-protected exit of the kiosk.

Caution : Configuring only a WorkSpace does not secure the terminal: with a WorkSpace only, all terminal functions (system, Android) are accessible to the user. If you wish to offer your users a semi-open, controlled mode, we strongly recommend that you also apply restrictions to the terminal via a security profile (see 12.2 page 51).

6.1. Designing a WorkSpace



6.1.1. Adding items to the WorkSpace

Device inventory
You can select a device to view its Software Inventory and give access to one of the applications in the WorkSpace. The application must be installed on the device. It will not be installed by CLYD. If it is not present on the device, it will not be displayed in the device's WorkSpace.

Widget
You can grant access to various widgets
Wi-Fi config: Connect to a Wi-Fi by entering the password.
Wi-Fi: Activate or deactivate
Bluetooth: Pair the device with another equipment
Cellular data: Activate or deactivate
Brightness: Adjust brightness
Telephone: Grant access to the telephone function
Message: Have a message containing variables.
Sound: Adjust the volume + -
Locate: Activate or deactivate
Auto-rotate: Activate or deactivate
Connection: Make the device connect to the Clyd server
Flashlight: Activate or deactivate

Link to a website
You can grant access to a website by specifying a URL

Add a command (see 13.3 page 58)
Use to add commands and intents.

Add a group
Use to create a group where you can place up to 16 icons (applications, files)

Catalog
You can select applications or files from your catalog. Applications added to the WorkSpace will be installed on the device after an installation request is sent to the Google servers (enterprise store applications) or after running the APK file (in-house applications).
An application or a file can be downloaded in this menu; they will be loaded in the catalog and immediately available to be saved to your WorkSpace.

Apart from "File" items, the label and icon of each item on the WorkSpace can be customized using the pencil icon overlapping the item. This customization will of course be reflected in the WorkSpace displayed on the devices.

6.1.2. Configuring the WorkSpace

Create pages

You can configure tabs to distribute your applications/files/widgets, etc. across multiple pages.

Screen resolution

You can adapt the WorkSpace to the resolution of your devices, either by indicating a resolution or by selecting a device.

Screen orientation

Portrait/landscape mode for your WorkSpace design.

Style settings

You can add wallpaper, select a background color for the kiosk and select a color for apps names and title text.

Priority
Setting priority levels on your WorkSpaces allows CLYD to make the highest priority WorkSpace available the event that multiple WorkSpaces are deployed on a device (in the case of non-disjoint targets, for instance).
The highest priority is 1, the lowest priority is 9 and the default priority is 5.
If multiple WorkSpaces deployed on a device have the same priority level, the last deployed WorkSpace will be the one available on the device.
Examples:

- Three WorkSpaces are deployed on the same device. WorkSpace A has priority 6, WorkSpace B has priority 2 and WorkSpace C has priority 9 →WorkSpace B is available because it has the highest priority.
- Three WorkSpaces are deployed on the same device. WorkSpace A has priority 6, WorkSpace B has priority 6 and WorkSpace C has priority 9 (WorkSpace C is available because it has the highest priority.
- Three WorkSpaces are deployed on the same device. WorkSpace A has priority 2, WorkSpace B has priority 2 and WorkSpace C has priority 9 (the most recently deployed of WorkSpaces A and B is available.

Advanced properties

❖CLYD Browser
Using the built-in CLYD browser, which allows you to specify a particular behavior, for example, to lock the address bar.
❖You can choose to position elements (applications, widgets, commands, etc.) in your kiosk manually or automatically.

6.2. TAB: Security

Security elements are now defined as a full-fledged provisioning profile from the "'Provisioning'" menu. See chapter "Provisioning profiles → Security profiles" (see 12.2 page 51).

The "Security" tab no longer exists for new Workspaces since Clyd 6.1.0 and will disappear for existing Workspaces. This is why it is advisable to create a security provisioning profile using the elements defined in this tab and targeting the same endpoints. This profile creation can be done by clicking on the "Create a security profile" button. The creation of this security profile uses the security parameters, targets and distribution schedule of the current Workspace.

Please note that the security provisioning profile only applies to devices with Clyd DPC 6.1 and the ClydMediaContact 6.12 client at least. It is therefore important to update your fleet of devices and then distribute your security profile.

Once these operations have been carried out, it is advisable to remove the security parameters present in the Workspace by clicking on the "Delete" button because they will no longer be effective.

Note that in the event of a conflict between the security settings defined in the WorkSpace security tab and the security profile in the "Provisionning" menu, the settings in the provisionning security profiles always take precedence over those defined in the security tab.

6.3. Deploying a WorkSpace

If you want to save your WorkSpace without broadcasting it, you can click on "Save Vx draft". You can also duplicate your WorkSpace in the WorkSpace list.

You cannot deploy if you have not selected a target. When you add targets in the TARGETS tab, the "Deploy Vx" button will be available.

Immediate deployment.

This is the default deployment mode.
Click on the "Deploy Vx" button to deploy version x of your WorkSpace. If the devices have a Google-ID, the WorkSpace will be deployed immediately, otherwise it will be deployed the next time the device connects to the CLYD server (connection schedule).

Catalog applications (excluding inventory applications) will be copied and installed on the device when the WorkSpace is deployed.
Catalog files will be copied and installed on the device when the WorkSpace is deployed.

Scheduled deployment.

You can define a validity period in the Deployment Schedule tab. The Google-ID is not used and the WorkSpace will be deployed to devices that connect during the configured validity period. The applications will be copied and installed when the WorkSpace is deployed. The files will be copied during the WorkSpace deployment.

Example: validity period, starting today, from 22:00 to 06:00 every day except Saturday and Sunday.

The devices connect to the CLYD server according to the connection schedule, therefore a connection schedule must be configured to ensure the devices connect during this period.

Stopping deployment.

Whether the deployment in progress is manual or scheduled, it can be stopped at any time by clicking on the "Stop Vx deployment" button.

If you are using managed configurations, deploying a WorkSpace has the effect of publishing the selected managed configurations for the applications in your kiosk to all devices in the targets associated with the WorkSpace.

6.4. File synchronization

The catalog files saved to the WorkSpace will be copied when the WorkSpace is deployed. You can update these files automatically. You just need to update the file in the catalog and select a synchronization mode in the deployment schedule.

Synchronizing files "at each communication"
All files saved to the WorkSpace will be synchronized each time the device connects to the CLYD server.

Synchronizing files "at the first communication in the validity period"
All files saved to the WorkSpace will be synchronized when device connects during the validity period. If the device connects multiple times during the validity period, the files will only be synchronized once (the first time).

6.5. Updating an application

To update an application, you need to deploy a new WorkSpace with the new version of the application.
You can either modify your current WorkSpace (it will stop the deployment in progress), or duplicate it to keep the original version. Duplicating a WorkSpace allows you to test the new application on a restricted target.

In your catalog or directly in the WorkSpace design tab in the Catalog menu, download the new version and save it to the WorkSpace before deploying.

6.6. Deleting the WorkSpace from the device

A WorkSpace will be deleted from a device in the following situations:

Deleting a WorkSpace: You can delete one or more WorkSpaces from the WorkSpace list. The WorkSpace will be removed from the device the next time it communicates with the CLYD server.

Deleting a target: You can delete the target containing your device in the WorkSpace. The WorkSpace will be deleted from the device at the next deployment (A delete request will be broadcast to the device).

Deleting the device in a target: If you delete your device from a target that is used in a WorkSpace, the WorkSpace will be deleted from the device at the next broadcast (A delete request will be broadcast to the device).

Warning: deleting a device from a company or unassigning a device does not remove the WorkSpace from this device. You need to uninstall the CLYDMediaContact client from the device or reset the CLYDMediaContact client.

Chapter 7. I want to master semi-opened and unlocked devices (EMM Profiles)

Semi-open terminal management allows free access to the system and then the desired restrictions to be applied in order to control terminals. ***The "EMM profile" node, which used to enable this, is to be deprecated****, but will continue to function for those already created.*
Since Clyd 6.3.0, you must use security profiles and WorkSpace to control semi-open terminals (see 12.2 on page 52).

Caution, in the event of a conflict between the EMM profile and the security profile, the security profile parameters always take precedence over those of the EMM profile.

Applicable management policies:

[Work Device] tab

Deploying in-house applications and the Google Play Store application:
You can select the applications to be installed on the device when the profile is deployed. To update an in-house application, the new version must be selected and the profile must be redeployed.

Deploying files:
You can select the files to be copied to the device when the profile is deployed. In the deployment schedule, you can choose to synchronize these files each time the device communicates or only when the profile is deployed.

Managing access to the Google Play Store:
You can control the device user's access to Google Play Store.

- Only to compulsory applications
- To some Enterprise Store applications and to compulsory applications
- To all Enterprise Store applications
- To all applications

[Security] tab

See chapter "Provisioning profiles → Security profiles" (12.2 page 51)

Security elements are now defined as a full-fledged provisioning profile from the "'Provisioning'" menu.

The "Security" tab no longer exists for new kiosks since Clyd 6.1.0 and will disappear for existing kiosks. This is why it is advisable to create a security provisioning profile using the elements defined in this tab and targeting the same endpoints. This profile creation can be done by clicking on the "Create a security profile" button. The creation of this security profile uses the security parameters, targets and distribution schedule of the current kiosk.

Please note that the security provisioning profile only applies to devices with Clyd DPC 6.1 and the ClydMediaContact 6.12 client at least. It is therefore important to update your fleet of devices and then distribute your security profile.

Once these operations have been carried out, it is advisable to remove the security parameters present in the kiosks by clicking on the "Delete" button because they will no longer be effective.

Note that in the event of a conflict between the security settings defined in the WorkSpace security tab and the security profile in the "Provisionning" menu, the settings in the provisionning security profiles always take precedence over those defined in the security tab.

7.1. Deploying an EMM profile

If you want to save your EMM profile without deploying it, click on "Save Vx draft" in the EMM profile list.

You cannot deploy if you have not selected a target. When you add targets in the TARGETS tab, the "Deploy Vx" button will be available.

Immediate deployment.

This is the default deployment mode.
Click on the "Deploy Vx" button to deploy version x of your EMM profile. If the devices have a Google-ID, the EMM profile will be deployed immediately, otherwise it will be deployed the next time the device connects to the CLYD server (see 15.1).

Catalog applications (excluding inventory applications) will be copied and installed on the device when the EMM profile is deployed.
Catalog files will be copied and installed on the device when the EMM profile is deployed.

Scheduled deployment.

You can define a validity period in the Deployment Schedule tab. Google-ID is not used. The EMM profile will be deployed to devices that connect during the configured validity period. The applications will be copied and installed when the EMM profile is deployed. The files will be copied during the EMM profile deployment.

Example: validity period, starting today, from 22:00 to 06:00 every day except Saturday and Sunday.

The devices connect to the CLYD server according to the connection schedule, therefore a connection schedule must be configured to ensure the devices connect during this period.

Stopping deployment.

Whether the deployment in progress is manual or scheduled, it can be stopped at any time by clicking on the "Stop Vx deployment" button.

If you are using managed configurations, deploying an EMM profile has the effect of publishing the selected managed configurations for the required applications in your EMM profile to all devices in the targets associated with the EMM profile.

7.2. File synchronization

The catalog files saved to the EMM profile will be copied when it is deployed. You can update these files automatically. You just need to update the file in the catalog and select a synchronization mode in the deployment schedule.

Synchronizing files "at each communication"
All files saved to the WorkSpace will be synchronized each time the device connects to the CLYD server.

Synchronizing files "at the first communication in the validity period"
All files saved to the EMM profile will be synchronized when the device connects during the validity period. If the device connects multiple times during the validity period, the files will only be synchronized once (the first time).

7.3. Updating an application

To update an application, you need to deploy a new EMM profile with the new version of the application.
You can either modify your current EMM profile (it will stop the deployment in progress), or duplicate it to keep the original version. Duplicating a EMM profile allows you to test the new application on a restricted target.

In the catalog or directly in the EMM profile design tab in the Catalog menu, download the new version and save it to the kiosk before deploying.

7.4. Deleting the EMM profile on the device

An EMM profile will be deleted from a device in the following situations:

Deleting an EMM profile: You can delete one or more EMM profiles from the EMM profile list. The EMM profile will be removed from the device the next time it communicates with the CLYD server.

Deleting a target: You can delete the target containing your device in the EMM profile. The EMM profile will be deleted from the device at the next deployment (A delete request will be broadcast to the device).

Deleting the device in a target: If you delete your device from a target that is used in an EMM profile, the EMM profile will be deleted from the device at the next deployment (A delete request will be broadcast to the device).

Warning: deleting a device from a company or unassigning a device does not remove the EMM profile from this device. You must reset the device to factory settings.

Chapter 8. I want to check the status of my fleet

Click on the "Devices" menu/tile to access the device list.

You can choose the number of devices to be displayed per page, view the number of items displayed on the current page, view the total number of pages, navigate from one page to the next, or directly view the items on the last page.
You can sort items by clicking on the title of one of the columns (ascending/descending).

A select function is available to add devices or apps data to be displayed in this view (which then appear as new columns in the list view).

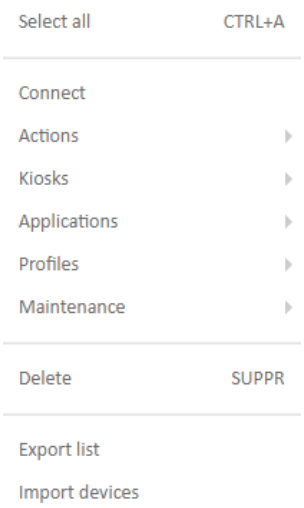
A Filter function is available (Show/hide filter areas), which allows you to manage the devices displayed. The filter remains active for the duration of the current session.

A geolocation function is available (Show geolocation), which displays a map with the location of your devices. A filter can be applied to limit the number of devices displayed on the map.

You can click on the name or identifier of a device to access its information sheet.

8.1. I want to take specific actions on my devices

A hamburger menu exists for specific actions on devices.
Actions are grouped into categories using a system of sub-menus.
The "Unassign", "Delete" and "Import devices" actions apply to the device directory (no call to the device).
All other action requests are sent to the device with the Google-ID via an FCM call (requires Internet access).



Connect: Request the device to connect to the Clyd server. The server takes advantage of this connection request to send back to Google servers the information related to the applications that must be present on the device. Since Clyd 6.1.0, the server no longer prepares the synchronization data for the device concerned (Product Policy).
Delete: Delete the device from the company (**caution**: this action is not reversible without action on the device)
***Import devices*:** Import devices (creation and/or modification) from a CSV file.
Export the list: creates a CSV export of the displayed list, including the displayed columns.

"Actions" category:

Request an inventory: Request a new inventory from the device.
Geolocalize: Request the device GPS coordinates.
Caution, to geolocalize your device, be sure to be compliant with your local rules (GDPR in Europe).
Broadcast a message: Broadcast a message to the devices.
Ring: Use to activate the ringtone of a device.

"Kiosks" category:

Start the kiosk: Start or stop the kiosk on the device.
Stop the kiosk: Stop the kiosk on the device.

Collect blocked windows: sends an FCM message to the MediaContact client to retrieve traces of the device's last 500 blocked windows in a "txt" file (the device must have a Google ID). Multiple devices cannot be selected for this action.

"Application" caregory:

Enable an application: Activate an application to make it usable.
Disable an application: Deactivate an application to make it no longer usable.
Uninstall an application: Uninstall a non-system application from the device.
Retrieve reports: Retrieve the reports generated by the Play Store applications installed on this device to add them to the device logs in Clyd. Only reports already retrieved by Google servers will be retrieved and displayed in the Clyd console.
Force report upload: Ask Google servers to retrieve the latest reports generated by the Play Store applications on this device but not yet available on Google servers. Please note that this action can only be performed 3 times per day and per device.

"Monitoring" category:

Apply profiles: Apply a specific monitoring profile to a device.

"Security" category:

Pause profile: *Pause the terminal's current security profile.*
Caution: In this case, all restrictions on the terminal are lifted, including those on the CLYDMediaContact client. There is no effect on system update settings, Playstore access or application permissions. It is necessary to reapply the profile once the actions have been carried out, otherwise no further restrictions are applied to the terminal.
Reapply profile: reapplies the terminal's current security profile.

"Maintenance" category:

Reboot: Reboot the device.
Reset (Wipe): Reset to device factory values.
Reinitialize (MediaContact) : Reinitialize the MediaContact client remotely (the device must have a Google ID, as the request is made via an FCM call).
Reload: Send all its Clyd configuration back to the device. The server takes advantage of this connection request to send back to Google servers the information related to the applications that must be present on the device (Product Policy).
Lock: Lock the device with a password.
Unlock: Remove the locking password.
Unassign (advanced):
When deallocating an EMM device, we delete the GOOGLE IDs (EMM Device ID and EMM User ID).
Caution: Following deallocation, the device can no longer connect to the CLYD server; you must reinitialize the CLYDMediaContact client on the device.
We keep the devices information (logs, software and hardware inventories).
Delete a device (advanced):
All device information is deleted (logs, hardware and software inventories GOOGLE IDs, EMM Device ID and EMM User ID).
Caution: After deletion, no further interaction with the deleted device is possible without a complete factory reset on the device (wipe) and a new enrolment.

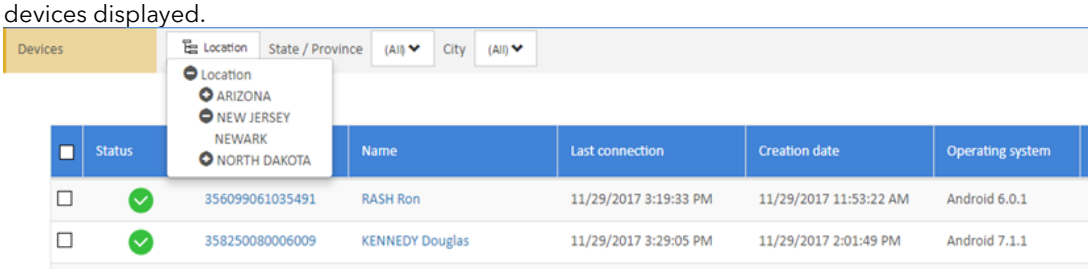
8.2. I want to create specific views (advanced)

A view function is available (view management), enabling you to display a company's devices by grouping them according to one or more criterias. Note that it is possible to give specific rights on views (see 15.3.7 page 76)

A view function is available (View management), which displays company devices grouped according to one or more criteria.

To create a new view, click on the "View management" icon, then the "Add view" submenu. Select the criteria that will define your views. Add criteria by clicking on "+"

Once created, the view is available in the icon bar. Then you can simply select the level(s) required to filter the devices displayed.



The view can be used to group company devices according to one or more criteria in order to facilitate user access (Example: grouping by department, by geographic area, by device type, etc.). A view belongs to one and only one company.

You can define access rights (Operator or user) at view level. For example, you can give Operator rights to an account at the "ARIZONA" view level to allow the operator to manage the devices in that region.

8.3. I want to import/modify devices (advanced)

The action menu offers the "Import devices" function which allows you to add new devices before they are enrolled or to modify existing devices from a CSV file (UTF-8).

When you select this menu, a window appears allowing you to select:

- The type of import you want to perform;
- The CSV file that contains the data to import into the Clyd directory.

Import type:
Modification : only the devices present in the CSV file and already existing in the Clyd directory will be processed. They will be modified with the values provided in the CSV file. The devices present in the CSV file and absent from the Clyd directory will be ignored.

Creation: only the devices present in the CSV file and absent from the Clyd directory will be processed. They will be added with the values provided in the CSV file. The devices present in the CSV file and present in the Clyd directory will be ignored.

Creation and modification: it is the combination of both other types of import.

CSV file:
The CSV file must respect a certain number of rules to be correctly interpreted by Clyd:

- Accepted separators: semicolon, comma, pipe
- Accepted delimiter : double quote
- Each line must be composed of the same number of fields as the number of columns defined in the file header.

Column names and order to import	Clyd columns
Identifier Name	Identifier (compulsory) Name

FirstName	Firstname
Street	Address
PostCode	Zip code
City	City
Country	Country
State	State/Province
UserFunction	Function
Service	Service
Location	Location
FixedPhone	Landline telephone
MobilePhone	Cellphone
Email	Email
Var:xxx	Value of the variable named xxx

The association between the CSV file and the Clyd directory is made on the mandatory field "Identifier".

Exécution report:
After the execution, a report is displayed indicating:

- Major errors preventing the import from running:
  - Incorrect file format
  - Duplicate column name
  - Non-existent "Identifier" column
- Otherwise, the number of modified, added, ignored and failed devices.

The following cases are ignored and the import is executed:

- Unknown column name (not corresponding to any expected field)
- For a column referring to a variable, unknown variable name
- Empty column name

When executing the import of a row in the CSV file:

- The following cases are not counted as errors:
  - The row is empty.
  - The fields present in the CSV file but not valued (empty string) are emptied in the imported device details.
- The following cases are counted as errors, the row is not imported and the execution continues with the next row:
  - Number of fields in the row different from the number of columns in the header row
  - Invalid data in a field: length, type
  - "Email" with an incorrect email value
  - Empty "Identifier" field
  - Empty "Name" field
  - Variable value not consistent with the variable definition

In case of import in addition, if the "Name" column is missing in the CSV file, the import assigns the value of the identifier to the "Name" field in the device details.

For variables with multiple values, only one value is possible.

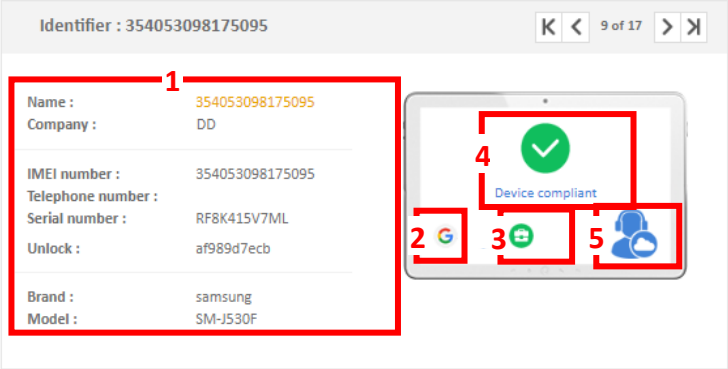
Chapter 9. I want to check the details of a specific device

From the list of my devices ("Devices" menu), I can view detailed information about a device by clicking on its identifier or name.

A menu, at the top-right corner of the page allows you to:

- To trigger actions on your devices, these actions are the same as those available from the "Devices" menu (see 8.1 before).
- Refresh the page to update the device data (an update of IPv4 WIFI, SSID, batteries, IMSI, ICCID, phone number is requested by FCM when opening this form). Close the device form and return to the view of all devices ("Devices").
- Close the device file and return to the view of all devices ("Devices").

9.1. I want to check the information of a specific device



An insert (1) shows identifying information about the device, including the name given and the Clyd company. In addition to the serial number or IMEI number, you can identify the device's phone number (from Android 13), brand or model.

(2) This icon allows you to verify that the device has the Google services

(3) This icon allows you to verify that the device is enrolled in Android Enterprise mode, binded to your company with Google.

(4) This icon shows the compliance of the device (or noncompliance) with the criteria you define in Clyd (see 15.6 page 78)

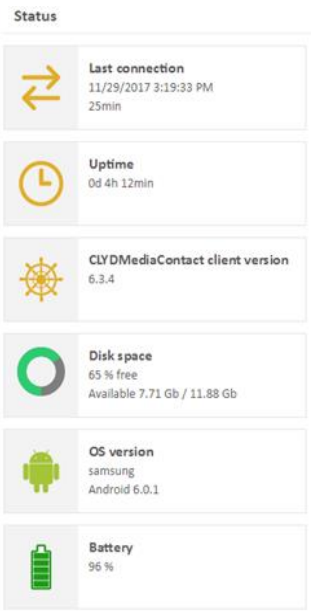
(5) Allows you to take remote control of your device (see 9.2 below)

Data update:

A request to update certain hardware information (battery, disk space, phone number, SSID, IMSI, IPv4 Wifi, ICCID) is made by FCM call when the device infos is opened. To update the information displayed, use the "update" button.

In all cases, all hardware information is updated during the inventory schedule (see 15.2 page 73) and transmitted to the server at the next call.

Status



Last connection
Date and time of the last connection of the device to the CLYD server

CLYDMediaContact client version
CLYDMediaContact client version installed on the device

OS Version
Device brand and Android version

Uptime
Date and time the device was last restarted

Disk space
Available disk space on the device

Battery
Device battery level

These informations (Battery, Disk space) can also be updated when an inventory request is made via the Hamburger menu on the device information sheet.

My device

General
Provides access to the device information sheet. This information can be
edited by an administrator or an operator.

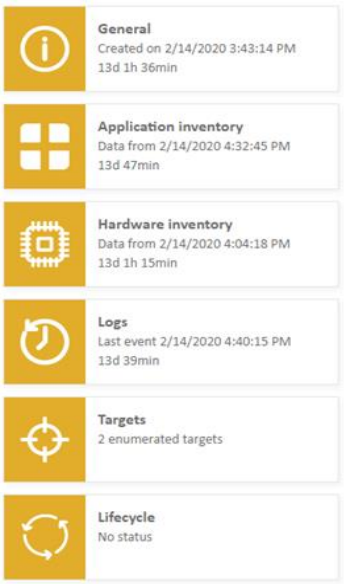
Application inventory
Applications installed on the device.
The system applications are filtered by default.

Hardware inventory
Device hardware information.

Logs
View all the actions performed on the device.

Targets
List of targets to which the device belongs.

Life cycle
Change the life cycle status of the device.



Functions

Kiosk or WorkSpace
View the status of your kiosk (“Started”, “Started, applications being installed”, “In process of deployment”, “Stopped”, “No kiosk”) or your WorkSpace (“Deployed”, “Deployed, applications being installed”, “No WorkSpace”)
Access the design of your kiosk or workspace.

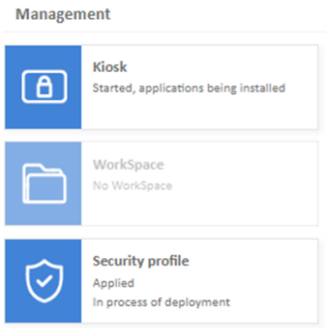
The status “Started, applications being installed” indicates that the kiosk is displayed but that some applications have not yet been installed (they are being downloaded from Google Play Store).
The same applies to workspace.

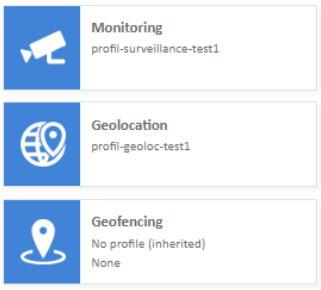
Security profile
View the status of security profile(s) applied to your device (no security profile, applied, in process of deployment) and access the configuration of your profile(s).

Monitoring
Access to your monitoring data (based on the monitoring profile).

Geolocation
Access to your device’s geolocation. Last known coordinates.

Geofencing
Access to a device’s geofencing data.





9.2. I want to take control of a device

Click on the “Remote control” logo and select remote control via MYCLOUD (Internet) or LAN (Intranet)

Remote control is not configured. To configure it, please go to the Configuration menu / Remote Control (or see 17.1 page 84).

Remote control is configured in LAN mode.

Remote control is configured in MYCLOUD mode, the device is not ONLINE.

Remote control is configured in MYCLOUD mode, the device is ONLINE.

Remote control requires the installation of the WiseMo Guest component which provides all the functionalities to control remotely, transfer files, etc. It can be downloaded from the CLYD server by clicking on the “Remote control” logo. It is an MSI package to be installed on the user workstation for the console. If you have a myCloud account, WiseMo Guest does not require a specific serial number to operate and you can use it in both myCloud and LAN modes.

If you have a myCloud account, WiseMo Guest does not require a specific serial number to operate, and you can use it in both myCloud and LAN mode.

Otherwise, in LAN mode, to take control from a device that is not exposed to the Internet, you'll need a perpetual license to use remote control (license to be entered in WiseMo Guest).

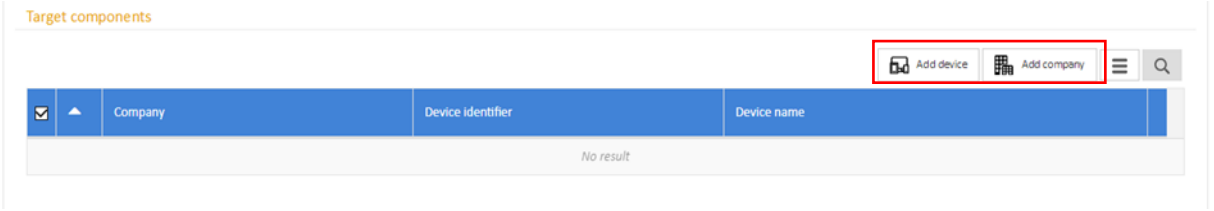
Chapter 10. I want to manage groups of devices

i Targets” make it possible to group a set of devices either by enumeration (enumerated targets) or by selecting criteria (targets with criteria). These targets can be used for connection schedules, for running processes, as well as for kiosks, WorkSpaces and provisioning profiles deployments.

10.1. Listed targets

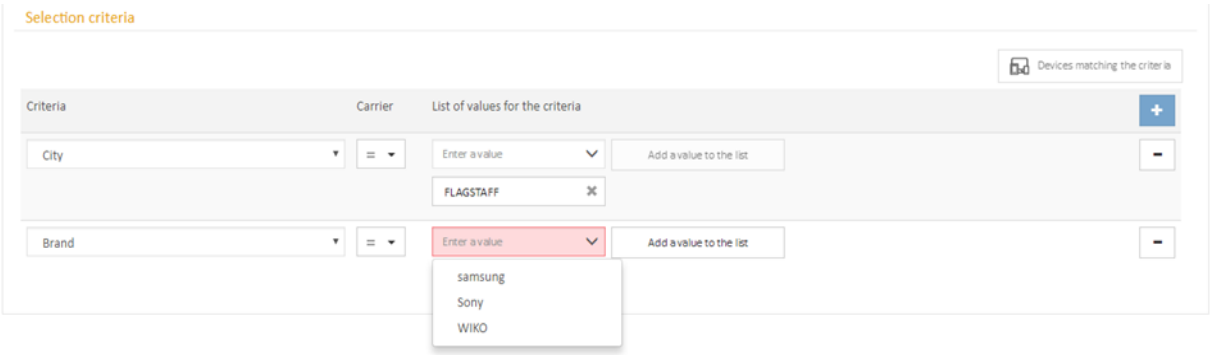
This type of target allows you to group one or more devices of one or more companies, depending on your administration level (Global Administrator or Company Administrator).

You can also add a company to this type of target. In this case, each time a device is registered with the company, it will automatically be included in the target. For example, if you deploy a kiosk with a target containing your company, when you register a new device with this company the kiosk will automatically be deployed to this device.



10.2. Targets with criteria

This type of target allows you to group devices according to certain criteria. The criteria are the device properties (brand, model, contact information, etc.) or in-house or system application versions (ExitCodeAndroid = 2.0, etc.). The values can be entered manually or selected from the available values in the CLYD database.



Warning: During the initial communication from a new device to the server, criteria on aliases or applications are not evaluated. Items associated with a target containing these criteria will not be downloaded to the device when it registers. They will have to be redeployed once the device's inventory data has been integrated on the server, which could take several minutes. However, all other criteria are taken into account from the initial communication: contact information (see the "Contact information" tab of the installation package) and information about device brand, model and operating system. A new device integrating a target based on these criteria will inherit the profiles already distributed for this target (kiosk, Workspace, profile EMM, process...).

10.3. Creating a target

Targets can be created or modified in the Targets menu, or when used (Connection schedule, Kiosks, WorkSpaces, Provisioning, Processes).

Chapter 11. I want to manage my application catalog

Use the catalog to create a repository of programs (in-house apps and Google Play Store apps) and files that will be used in CLYD (Kiosk, WorkSpace). In-house applications and files are uploaded to the CLYD server before use. Applications from the enterprise store must first be selected from the Google Play Store before being used.

You can access the catalog via the menu or the tile on the home screen.

As with other list-views, an action menu allows you select all items, delete, or export the list as displayed on the interface.

You can filter catalog items by clicking on the filter button and then filtering the relevant columns.

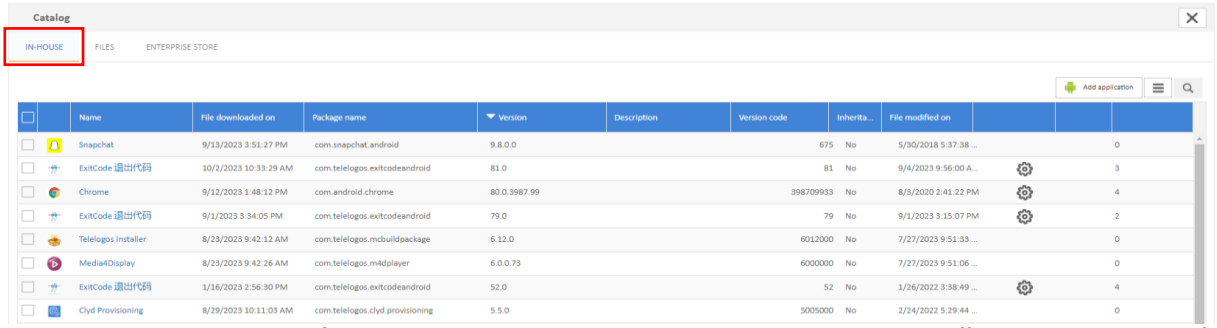
With the list-views, you can also :

- Sort the columns by clicking on their headers (one click changes the sort order).
- Change the column display order with drag & drop.

11.1. I want to deploy applications or files in my catalog

❖ ***In-house applications***

Add in-house applications (.apk) to your catalog:

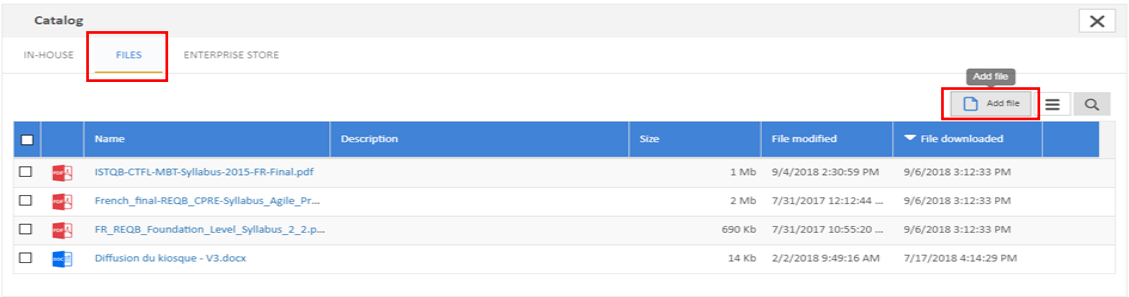


The "Propose managed configurations" column indicates whether the application offers the possibility of defining managed configurations (advanced - see 11.3.2 page 47).

The "Inheritance" column indicates whether the application has been added to the catalog by inheritance (advanced - see 11.6 page 50)

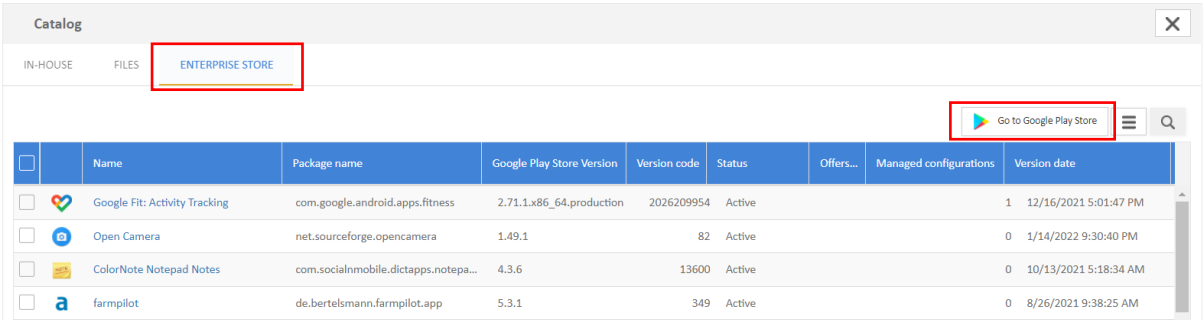
❖ ***Files***

Add files to your catalog:



Google Play Store applications (Android Enterprise Only)

Prerequisite: Company must be registered with Android Enterprise (see chapter Chapter 1 page 12) and apps must be sent to devices with Google ID.



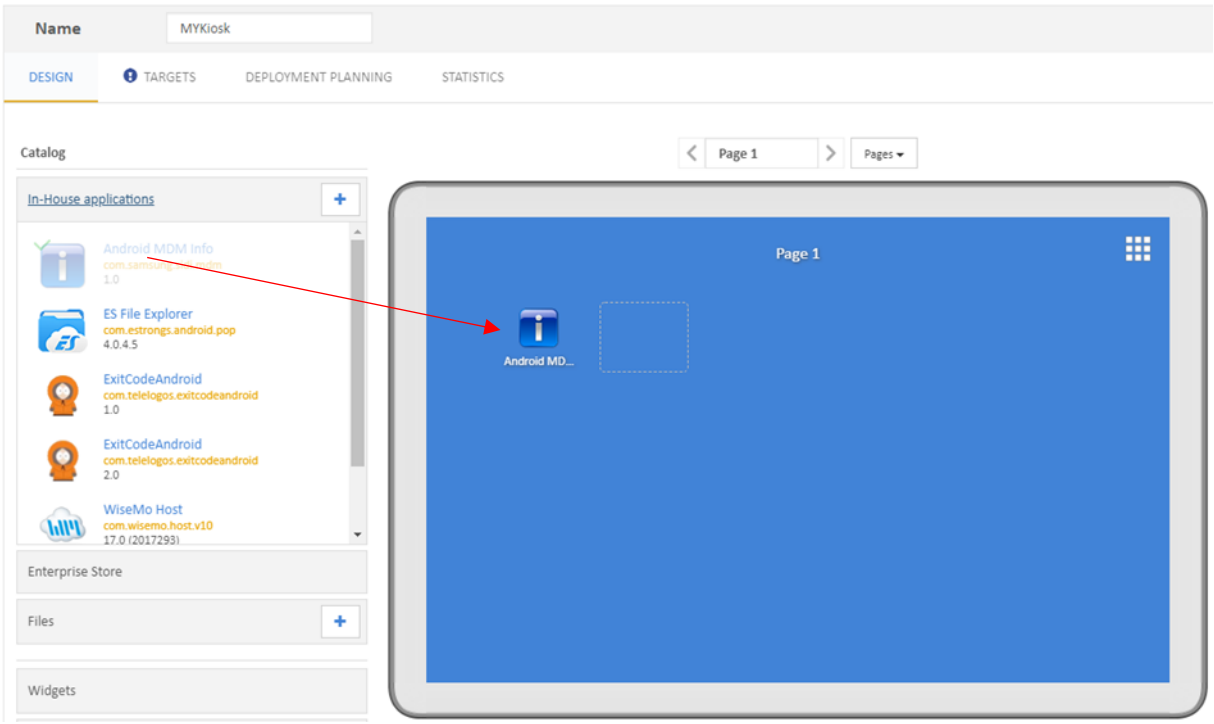
By clicking on "Go to Google Play Store", you can reference applications from the Google Play Store in your CLYD corporate store (or even web applications).

The 'Status' column contains 'Active' if the application is still available in the Play Store. It contains 'Removed from Play Store' if it is no longer available.

The cogwheel indicates that the application offers the managed configurations (see 11.3.2 page 47).

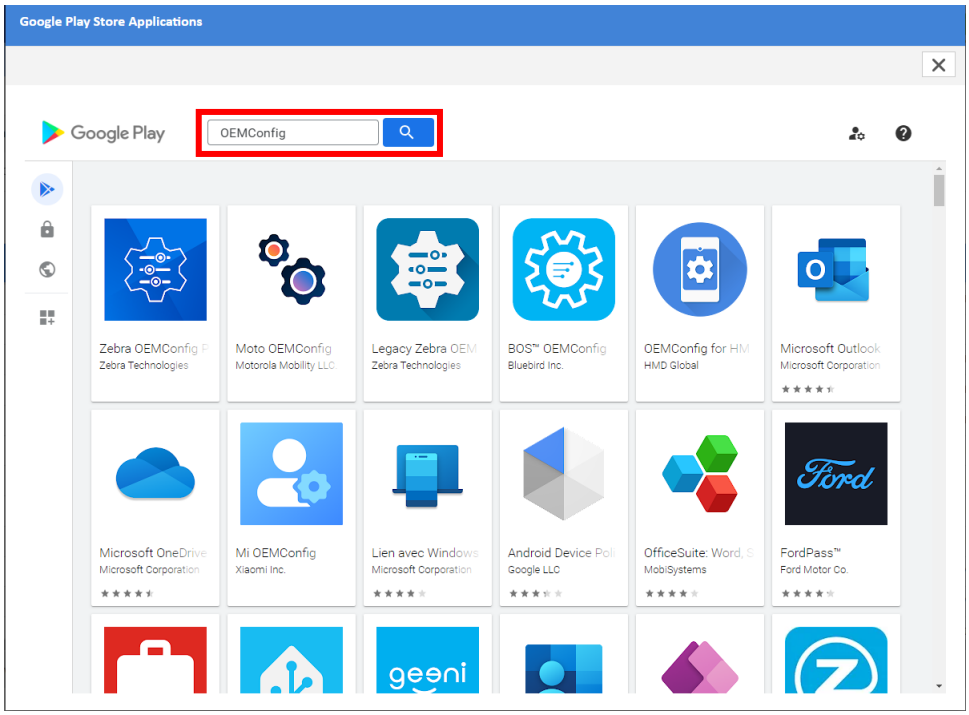
❖ ***Using the catalog***

You can access the catalog from the design interface of the kiosk and the WorkSpace. Double-click or drag and drop to place the applications/files on the kiosk or WorkSpace image.



11.2. I want to appply additional specific configurations to my devices (OEM Config)

In Clyd, you can use OEM Config to customize specific settings for Android terminals. Indeed, some terminal manufacturers allow you to manage additional parameters that you can handle. The settings available depend on what the OEM includes in its OEMConfig application.



To set OEMConfigs, go to the the Enterprise store. Access Google Play store, you can obtain the targeted OEM Config application and integrate it into your catalog by searching for "OEM Config" or "OEM" and the manufacturer of your terminals. If not, contact your device supplier to find out whether a compatible OEM application is available.

11.3. I want to manage additional settings for my applications

From the catalog, by clicking on their names, you can view the details of files and applications and manage additional parameters:

- In the "General" tab: define a description for applications and files.
- For Enterprise Store applications, in the "Update mode" tab: you can manage the update mode that Google will perform for the application (see below, 11.3.1 page 46)
- For applications that allow it, in the "Managed configurations" tab: create, modify or delete managed configurations (see below, 11.3.2 page 47).

11.3.1. I want to choose the update mode for my enterprise store applications (advanced)

Prerequisite: These settings can only be managed for applications from the enterprise store ("enterprise store" tab in the Catalog).

At any time, you can choose the update mode of an application from the Google Playstore (by clicking on the application in the enterprise store, then on the "update mode" tab). The update mode of an enterprise store application tells the Google servers which rule to apply to update this application on the devices:

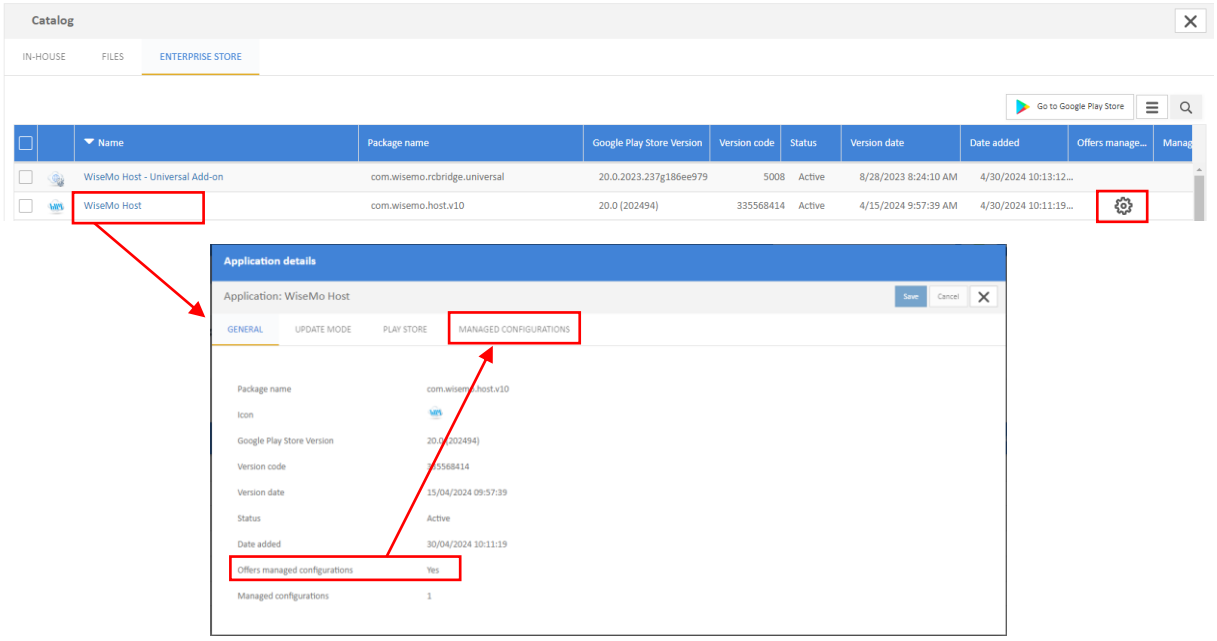
- High priority: the update is pushed as soon as it is available. In this case, the usual rules of updating are no longer valid, the update will be done even if the device is not connected to Wi-Fi, even if it is not plugged in and even if the application is in the foreground.
- Default: this is the usual mode, which requires the device to be plugged in, inactive, connected to Wi-Fi and the application is not running in the foreground. The condition on Wi-Fi can be relaxed by changing the application update policy configurable in the Security tab of kiosks, WorkSpaces and EMM profiles.
- Update postponed for 90 days after publication on the Play Store.

11.3.2. I want to set managed configurations for my applications (advanced)

By clicking on the name of an application (or from the catalog), you can check whether this application offers managed configurations and, if so, identify the number of configurations already created on Clyd. This information is visible from the catalog.

> **Caution,** This feature is new in Clyd version 6.2 for In-House applications. You need to check that configurations exist for applications that previously existed in your catalog. This check is automatically triggered by the server when you open your application details (additional parameters).

This functionality is for applications that have been designed to be remotely configurable (e.g. OEMConfig applications provided by device manufacturers). The managed configurations of an application can be accessed by opening the details of this application in the enterprise store of the CLYD catalog using "Managed configurations" tab.



Managed configurations can be accessed by clicking on the application name in the enterprise store and are for applications in the Google Play Store where you can enter their settings. These parameters can be values or references to device information (variables or aliases):

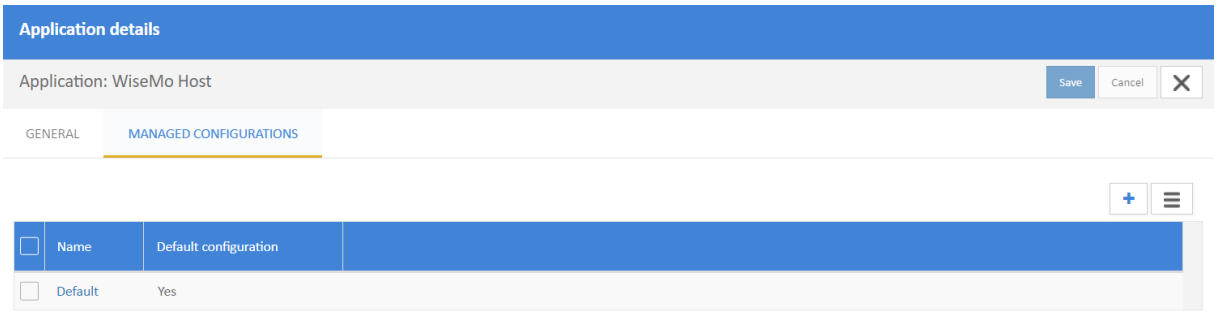
%USER%: Device identifier
Device coordinates:
%T_USERS.Name%, %T_USERS.Street%, %T_USERS.City%, %T_USERS.PostCode%, %T_USERS.Country%
%T_USERS.UserFunction%, %T_USERS.Service%, %T_USERS.Localisation%
%DATE%
%AliasStation=NAME_Alias%
…

If you use CLYDMediaContact variables or aliases, they will be assigned each time the managed configuration is published before it is pushed to the devices in your device population.

Multiple managed configurations can be defined to apply different application settings for specific devices. This targeting is done by using kiosks, WorkSpaces and EMM profiles ("Compulsory applications" section). When adding an application with managed configurations to a kiosk, WorkSpace or EMM profile, you can associate this application with any of its managed configurations, or with the default managed configuration. Each time the kiosk, WorkSpace or EMM profile is deployed, the targeted devices will receive the managed configuration selected for each of the applications present in the kiosk, WorkSpace or EMM profile.

In In-House mode, when you add a new version of an apk to the catalog, you inherit the managed configurations created for the most recent version used on Clyd. You can modify or delete these configurations and add new ones. If this new version of your apk proposes new parameters via its managed configurations, you must modify

the inherited managed configurations and define these new parameters (otherwise they retain their default values).



- **Editing and deleting a managed configuration.**

Save:

Application list: [Click on the application name]
Managed Configurations tab: [Edit from list]
[Save] button in the edit window

You can edit fields and/or modify options. If you save, the managed configuration will not be published on the devices that already have the application. Configurations are published when a kiosk, WorkSpace or EMM profile is deployed that contains applications associated with a managed configuration.

When editing and saving an existing managed configuration:

- It is not published: no change on the devices already deployed.
- Devices retrieve the managed configuration when the kiosk, WorkSpace or EMM profile using that managed configuration is published.

Delete:

Application list: [Click on the application name]
Managed Configurations tab: [Select from list]
Hamburger menu: [Delete]

You can delete a managed configuration from the list of managed configurations for an application only if this managed configuration isn't associated to any kiosk, WorkSpace or EMM profile.

Duplicate:

List of applications: [Click on the name of the application]
Managed Configurations Tab: [Edit from List]
[Duplicate] button in the edit window

Applications that are no longer available in the Google Play Store are not removed from the CLYD enterprise store. Their managed configurations are not deleted either. This means that when the application is available again in the Google Play Store, the previously created managed configurations will be usable again.

- **Setting a managed configuration as default.**

Application list: [Click on the application name]
Managed Configurations tab: [Select from list]
Hamburger menu: [Set as default configuration]

The default managed configuration of an application will be automatically associated when the application is added to a kiosk, WorkSpace or EMM profile.
If there is no default configuration for an application, then no managed configuration will be published on devices that receive a kiosk, WorkSpace or EMM profile containing that application and associated with the default managed configuration.

- **Removing a managed configuration as default.**

Application list: [Click on the application name]
Managed Configurations tab: [Select from list]
Hamburger menu: [Remove as default configuration]

The deactivation of this "Default configuration" attribute will impact kiosks, WorkSpaces and EMM profiles that use the default managed configuration **since the devices that have previously received theses kiosks,**

WorkSpaces and EMM profiles will lose their configuration for this application the next time the kiosk, WorkSpace of EMM profile is deployed.

- **Publish managed configurations.**

> Prerequisite: to receive managed configurations, devices must have ClydDPC installed and therefore have been enrolled on an Android Enterprise company or via an adb package (for admin devices).

Managed configurations are published each time a kiosk, WorkSpace or EMM profile containing the associated application is deployed. It is not possible to manually publish a managed configuration.

- **Initialize the device.**

When initializing a device, the MC client boots the device, deletes the Google EMM account if it exists and requests the creation of a new Google EMM account.
When the Google EMM account is created, the device sends its DeviceId via WebServices to the server.
When the server receives the DeviceId from a device, it requests the installation of EMM Apps if needed and publishes the associated managed configurations if they exist for this device only.

- **Choose the configuration to associate with an application (from a Kiosk / WorkSpace).**

To link a specific managed configuration to an application, from your Kiosk/Workspace/EMM Profile, point your cursor at the targeted application and click on the edit icon.

Deploying an application to a device by process does not publish a configuration to that device.

11.4. I want to understand distribution of a kiosk, a Workspace or an EMM profile with 1 or N applications from the Enterprise Store (advanced)

In-House applications are transferred and installed on the device by CLYD.
Applications from the Enterprise store are managed by the Google Play Store. When deploying the kiosk or the WorkSpace, CLYD asks the Google Play Store to install the applications on the devices of the target.
Reminder: (Only a device with a Google-ID):

- During deployment, CLYD asks Google to contact the devices so that they then call the CLYD server (This request uses the FCM protocol from Google - This request is kept for 28 days by Google).
- As soon as the device connects to the CLYD server, the kiosk or the WorkSpace, the files, the In-House applications are transferred and installed on the device, at the same time CLYD contacts the WEB services of the Google Play Store so that the applications of the store are installed on the device.
- Also during deployment, the managed configurations associated with the applications in the kiosk, WorkSpace or EMM profile are published.

In the case where the device does not have a Google-ID, or the device remains inaccessible for more than 28 days (FCM protocol delay), then the device will connect to the CLYD server according to the parameters of the connection schedule (default, connection every hour).

11.5. I want to understand how Playstore application installations works on a device initialization (advanced).

When initializing a device, MC client performs an initialization, deletes the Google EMM account if it exists, and requests the creation of a new Google EMM account.

When the Google EMM account is created, the device sends its DeviceId via WebServices to the server.

When the server receives a device's DeviceId, it requests the installation of EMM Apps if required, and publishes the associated managed configurations if they exist for this device only.

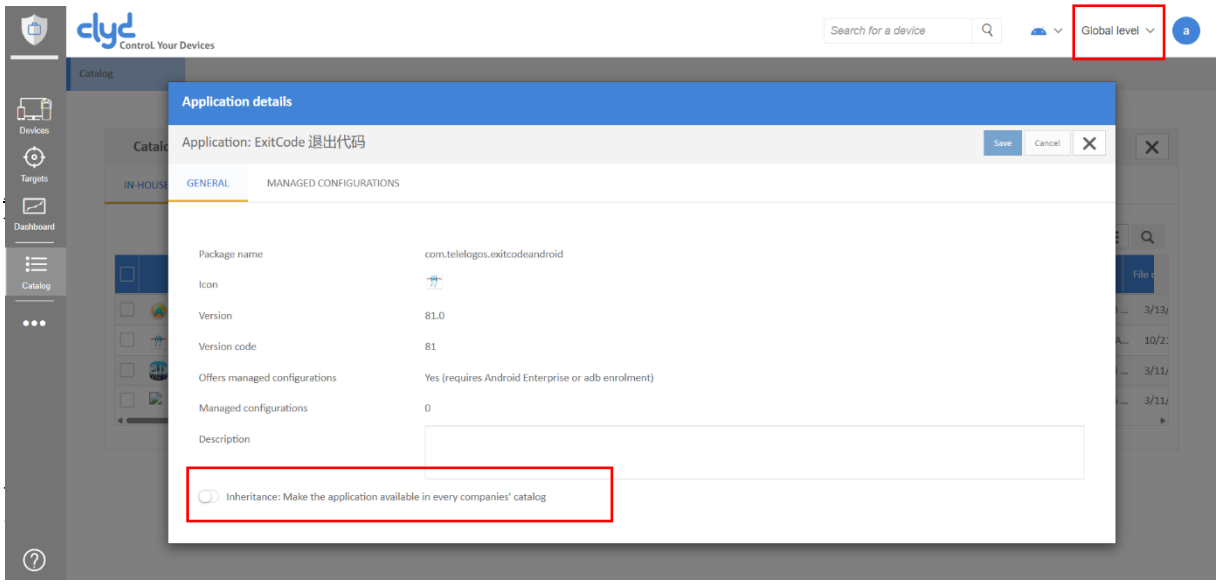
11.6. I want to set additional parameters on my server (advanced)

11.6.1. want to add applications to the catalog of all the companies on my server (advanced)

Prerequisite: be logged in with global administrator rights on your server (see 14.3 page 68)

At the global level, you can make your applications available to all the companies on your server. To do this, click on the application in question and then, in the application details, tick "Legacy: make application available to all companies". This application will then appear in your server's companies' catalog without any further action. You can remove this application from all company catalogs at any time by unchecking this option.

Caution: note that no checks are performed when an application is deleted at global level. If it is deployed in Kiosks, WorkSpace, EMM Profiles, installation packages or processes, it is not deleted from these profiles.



Chapter 12. I want to provide profiles to my devices (advanced)

12.1. Wi-Fi profile

The Wi-Fi provisioning profile allows to:

- Configure a Wi-Fi network on devices even if the network is not available at that time;
- Activate this Wi-Fi network on devices after configuring it.

The supported authentication types are the same as for the "prov_wifi" command (advanced, see 13.3):

- None (open networks)
- WEP (*note that WEP protocols have significant security issues and consider using WPA or WPA2 protocols*)
- WPA
- WPA2 Personal
- WPA2 Enterprise

In WPA2 Enterprise authentication, certificates may be required. Simply import them into the provisioning profile and Clyd will install them on the devices.

The Wi-Fi provisioning profile also allows you to define proxy and IP settings.

Multiple Wi-Fi profiles can be sent to the same device. In this case, each of the Wi-Fi networks present in the profiles will be created on this device.

12.2. Security Profiles

A security profile allows you to apply settings related to endpoint security.

Restrictions

A number of restrictions can be applied to the device in the following areas :

- Users/accounts (ex. Prohibit modification of accounts, etc.).)
- Applications (ex. Prohibit the installation of applications, etc..)
- Network (ex. Prohibit modification of Bluetooth configuration, etc.)
- System (eg Forbid Factory reset, etc.)

To be applied on your devices, you must target devices with minimum Android versions, as detailed in the table below.

Type	Restriction	Minimal Android version
User / account	Disallow_Add_User	Android 5.0
User / account	Disallow_Remove_User	Android 4.3.x
User / account	Disallow_User_Switch	Android 9
User / account	Disallow_Modify_Accounts	Android 4.3.x
Applications	Ensure_Verify_Apps	Android 5.0
Applications	Disallow_Install_Unknown_Sources	Android 4.3.x
Applications	Disallow_Apps_Control	Android 5.0
Network	Disallow_Airplane_Mode	Android 9
Network	Disallow_Config_Mobile_Network	Android 5.0
Network	Disallow_Config_Tethering	Android 5.0
Network	Disallow_Data_Roaming	Android 7.0
Network	Disallow_Network_reset	Android 6.0
Network	Disallow_Bluetooth	Android 8.0.0
Network	Disallow_Config_Bluetooth	Android 4.3.x
Network	Disallow_Config_VPN	Android 5.0
Network	Disallow_Config_Wifi	Android 4.3.x
Network	Disallow_Outgonig_Beam	Android 5.1
Network	Disallow_SMS	Android 5.0
Network	Disallow_Bluetooth_Sharing	Android 8.0

System	Disallow_Config_Date_Time	Android 9
System	Disallow_Config_Locale	Android 9
System	Disallow_Config_Location	Android 9
System	Disallow_Config_Screen_Timeout	Android 9
System	Disallow_Debbuging_Features	Android 5.0
System	Disallow_Mount_Physical_Media	Android 5.0
System	Disallow_System_Error_Dialogs	Android 9
System	Disallow_USB_File_Transfer	Android 4.3.x
System	Disallow_Safe_Boot	Android 6.0
System	Disallow_Factory_Reset	Android 5.0
System	Disallow_Share_Location	Android 4.3.x

Password Policy

- No change (default)
- Removedevice lock
- Imposing a password
- Specify rules

Standby

You can apply a parameter for standby (15s, 30s, 1min, 2mn, 5min, 10min, 30min, Never)

CLYDMediaContact Agent Security (Highly Recommended)

- Prohibit uninstallation of CLYDMediaContact client
- Imposing a password for access to options (modifying connection parameters, resetting the client). This option is necessary to pause security profile directly from CLYDMediaContact on the devices
- Make CLYDMediaContact client notifications non-clickable.

System update policy (Google forces system updates to apply)

- Manual
- Automatic (updates are installed as soon as available)
- Scheduled (set a time slot for applying updates)
- Postponed 30 days (except for security patches which cannot be postponed).

Note.

The system cannot install updates if the terminal is not connected, if disk space is insufficient or if the battery level is low.

To check locally on a device for available updates, go to your device settings / System / Software update.

Playstore management policy

- **Playstore access (apps access on the devices)**
  - "None" (only applications distributed via Kiosk and WorkSpace can be accessed from the Playstore)
  - "Some from the corporate store", in this case, select applications from the corporate catalog and add them to a list of applications that will then be accessible from the Playstore on the terminal side.
  - "To all applications in the enterprise store": in this case, the Playstore on your terminals provides access to the entire enterprise catalog managed on Clyd.
  - "To all applications", in which case the Playstore is completely open on the terminal side, and the user can freely add any new application of their choice to the terminal.
- **Playstore apps update policy**
  - Automatically, via any network
  - Automatically, via Wi-Fi only
  - Chosen by the user (in the device's Play Store application)
  - Never

Samsung specific settings (SDK KNOX)

- Forbid system updates
- Filter URLs: authorized domains, prohibited URLs (Android 9 to Android 13)
- Block buttons

Note. URL Filtering

- To deny device access to a particular URL:
  - Add the URL to be prohibited "*my_domain_forbidden.ext" to the list of prohibited URLs, by preceding the URL with a star
- • To restrict device access to a particular domain:

- Add "*" to the list of forbidden URLs
- Add the clyd server ("*my_domain_clyd.ext") in the authorized domains, by preceding the domain name with a star (otherwise communications will no longer be possible)
- Add the domain to be authorized "*my_domain_authorized.ext" in the authorized domains, by preceding the domain name with a star

Caution: be sure to allow access to Clyd and Android Enterprise servers in case of url filtering.

Permissions (management of Google Play Store App permissions)

- Global
- By application

Security profiles support the notion of precedence. Defining priority levels on your security profiles allows Clyd to apply the profile with the highest priority if several security profiles are deployed on a device (in the case of non-disjoint targets for example).
The highest priority is 1, the lowest priority is 9, and the default priority is 5.
In the event that several security profiles distributed on a device have the same level of priority, it is the profile distributed last that will be applied on the device.

12.3. Contacts profiles

The Contacts Provisioning Profile allows you to add, edit, delete, and/or favorite contacts on devices.

Contact provisioning is based on a CSV file that you need to import into Clyd.

- The file must have 4 or 7 fields, separated by semicolons (see expected file structure below).
- Each line in the file must contain the same number of fields. Some fields may be left blank (see examples below).
- The file must be saved in CSV format and encoded in UTF-8.

The structure of the file to be imported can take two different formats, depending on the information you wish to import or the actions you wish to take:

- 4-column format: "Last name;First name;Mobile phone;Email".
- 7-column format: "Last name;First name;Mobile phone;Email;Business phone;Service;Action" (with "Action" = "D" to delete the contact or "F" to bookmark it).

At provisioning time, if the contact does not exist, it is created. If the contact already exists (first name and surname), it is modified.

Examples

- **Import** contacts**:**

```
Thomas;Georges;0601020304;georges.th@azerty.com
Martin;Karen;;karen.martin@exemple.com
```

In this example, we don't know Karen's phone number (unlike Geroges'), so we leave the corresponding field empty.

- Add a contact to **favorites**:

```
Brown;Mary;0601020304;mary.brown@ex.com;;;F
Smith;Eric;;eric.smith@exemple.com;;;
Jones;Linda;;;;;
```

In this example, we add Mary as a favorite and we also add Eric and Linda without adding all their information.

- **Edit** an existing contact :

```
Smith;Eric;0706050403;;
```

In this example, if an "Eric Smith" contact existed, we modify the information known on the devices. Whatever information is known on the device, the only information set is its number.

Several contact profiles can be sent to the same device. In this case, orders to add, modify, delete and bookmark contacts will be executed on the device in the order in which the profiles are received.

You can download the latest version of a broadcast or saved contact profile.

Several contact profiles can be sent to the same device. In this case, the orders to add, modify, delete, and favorite contacts will be executed on the device in the order in which the profiles are received.

12.4. Deploying a provisioning profile

If you want to save your provisioning profile without deploying it, you can click on "Save Vx draft". You can also duplicate your provisioning profile in the provisioning profile list.

You cannot deploy if you have not selected a target. When you add targets in the TARGETS tab, the "Deploy Vx" button will be available.

Immediate deployment.

This is the default deployment mode.
Click on the "Deploy Vx" button to deploy version x of your provisioning profile. If the devices have a Google-ID, the provisioning profile will be deployed immediately, otherwise it will be deployed the next time the device connects to the CLYD server (connection schedule).

Scheduled deployment.

You can define a validity period in the Deployment Schedule tab. The Google-ID is not used and the provisioning profile will be deployed to devices that connect during the configured validity period.

Example: validity period, starting today, from 22:00 to 06:00 every day except Saturday and Sunday.

The devices connect to the CLYD server according to the connection schedule, therefore a connection schedule must be configured to ensure the devices connect during this period.

Stopping deployment.

Whether the deployment in progress is manual or scheduled, it can be stopped at any time by clicking on the "Stop Vx deployment" button.

12.5. Deleting a configuration

12.5.1. Wi-Fi Profiles

Deleting a Wi-Fi network previously configured by Clyd on a device is done by creating and executing a process using the command "prov_wifi -d {ssid}".

12.5.2. Security profiles

Deleting a security profile previously configured by Clyd on a device is done:

- By deleting the Clyd console side profile. The deletion will be taken into account on the device at the next communication. In addition, a connection request is sent to the device so that it connects as soon as possible (if the device has a GoogleId).
- By removing the targets containing this device from the profile. The deletion will be taken into account on the device at the next communication. In addition, a connection request is sent to the device so that it connects as soon as possible (if the device a GoogleId).
- By excluding this device from the targets associated with the profile. The deletion will be taken into account on the device the next time the profile is deployed.
- By deleting the targets associated with the profile and containing this device. The deletion will be taken into account on the device the next time the profile is deployed.

Caution: It is important to understand that deleting a security profile on the device does not reinitialize the security settings previously modified by this profile. Deleting the security profile on a device simply prevents future changes to that profile from taking effect.

To reinitialize the security parameters of a device, it is necessary to distribute a new security profile which cancels the modifications previously made on the device. For example, if Profile A has enforced a password policy on the device, Profile B will need to override the password settings on the device by choosing "Remove Device Lock".

12.5.1. Contacts profiles

Deleting all contacts previously provisioned by Clyd on a device is done by creating and executing a process using the "prov_contact -d" command.

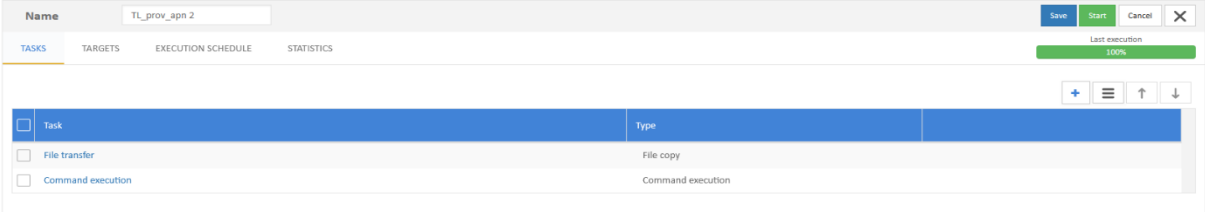
Contacts can also be deleted individually using the appropriate syntax in the contact provisioning file (via « Provisionning » menu).

- **Example**:

Dubois;Louise;;;;;D	In this example, if a "Louise Dubois" contact existed on the targeted device, we delete this contact from the device.

Chapter 13. I want to run processes on my devices (advanced)

Processes allow you to launch one or more actions on a device or set of devices. A process is composed of tasks such as updating the MediaContact client (Clyd agent), executing a system command, synchronizing or copying files, or distributing applications).



The burger menu of the process list allows you to duplicate one or more processes. Processes created by duplication are not executed automatically.

13.1. Adding tasks

Tasks allow actions to be performed on devices (executing commands, transferring files, distributing applications); these tasks are performed sequentially. You can change the task order, either by moving the task with your mouse or by selecting a task and using the arrows (up, down).

❖ **Execute a command (advanced, see 13.3 page 58)**
Use this task to execute a "pm ....." system command (caution: to execute system commands, the targeted devices must have a manufacturer-signed add-on - Samsung, Zebra and Sony are excluded.). You can execute CLYD commands, as well as intents.

❖ **File transfer (advanced)**
Use this task to copy or synchronize a file from the CLYD server to the device(s).

Operation: select transfer, copy or synchronize

Source: select an in-house file or application from the catalog

Destination: enter the destination path and filename. By default, the destination filename is the same as the source filename. MediaContact variables can be used in the destination path.

❖ **Distributing applications (advanced)**
Use this task to synchronize applications from the catalog to the devices and to install the applications either directly after the download or in deferred mode (installation can be deferred until the next device reboot or at a later date). Applications will be installed according to the transfer order.

Warning:
- A process can only contain one distribution task.
- A process-distributed application never embeds managed configurations.

Source: Select 1 or several in-house applications from your catalog. Applications will be installed on the device in the order of the selection.

Destination: Select a destination folder on your device for in-house applications.

Task properties:

Defer installation: By default, in-house applications are installed as soon as they are downloaded to the device. You can postpone the installation until the next device reboot or at a specified date.

Delete the file(s) after installation: By default, files are not deleted.

Upload the reports as soon as the installation is complete: When using deferred installations, the application installation reports are not uploaded in real time. They are delayed until the next device communication. This option allows you to report back just after installation. (The software inventory is uploaded in real time via web services)

❖ **Updating the CLYDMediaContact client**

This task updates the CLYD client. If your device is registered as an EMM (Android Enterprise) device, you can also update the CLYD DPC (Device Policy Controller). Since the DPC is on the Google Play Store, it should update itself automatically via Internet.

Warning: If the target contains both Android Enterprise and Device Admin devices, then the process will be seen as Not OK, but all devices will be updated. You can check the update by using a plug-in component.

13.2. Executing a process

Select a target:
You must add one or more targets containing the devices on which you want to execute the process tasks.

Select an execution schedule:

Immediate execution
Click on "Start" to trigger the process manually.

Start: The process is launched; each device will execute the tasks when it connects to the server. Devices with a GoogleID are called via FCM and connect to the CLYD server. Devices without a GoogleID connect to the CLYD server according to the communication schedule.

Stop: You can interrupt the process manually before it has been executed on all devices. By default, the process stops when all the target's devices have connected and executed the process.

Deferred execution
The process is launched according to a defined schedule. By default the process is not active. It must be started for the schedule to be taken into account.

Specific date:
Start the process so that it runs on a specific date. The process stops when all the target's devices have connected.
Stop allows you to interrupt the schedule or stop the current execution if the process is running.

Communication (at communication start-up):
Start the process so that it runs when the client is initialized.
Stop allows you to interrupt the schedule.

Hourly/daily frequency:
Start the process so that it runs according to the schedule.
Stop allows you to interrupt the schedule.

Stop execution:
Whether the execution in progress is manual or scheduled, it can be stopped at any time by clicking on the "Stop the process" button.

Execute in recovery mode:
The recovry mode allows the process to be replayed for the devices of the target for which the previous execution failed.

Statistics:
You can track process and task execution on each of the target's devices via the "Statistics" tab. A progress bar is displayed on the list of processes (click on the progress bar to go directly to the "Statistics" tab), as well as on the process sheet.
The statistics display the process currently being executed or the last time it was executed.

13.3. I want to trigger commands on my devices (advanced)

All commands (13.3.2) & intents (13.3.1) can be used in processes (task: command execution) and in a kiosk or WorkSpace (Add a command), as well as in an HTML page launched from the kiosk or WorkSpace.

13.3.1. Intents

Trigger the main application activity:
application://{com.name.package}

Example: Triggering the Calendar application
application://com.google.android.calendar

Trigger the main application activity with settings:
application://{com.name.package};launchFlags=0x10000000

Example: Triggering the Calendar application *without maintaining the status*
application://com.google.android.calendar;launchFlags=0x10008000
Example: Triggering the Calendar application *returning to the foreground and keeping its status*
application://com.google.android.calendar;launchFlags=0x00020000

Trigger message broadcast:
intent://#Intent;action={intent.action};{type}.{parameterName}={parameterValue};end

Example:
intent://#Intent;action=com.telelogos.exitcodeandroid.NOTIFY;S.data=Hello World!;end

Trigger an action:
intent://#Intent;action={intent.action};end

Example: Opening the Phone application
intent://#Intent;action=android.intent.action.DIAL;end

Trigger an action with parameters:
intent://{parameters}#Intent;action={intent.action};end

Example: Calling the number passed as a parameter
intent://tel:0241000000#Intent;action=android.intent.action.DIAL;end
Example: Preparing an SMS message
intent://smsto:12345678#Intent;action=android.intent.action.SENDTO;S.sms_body=The%20SMS%20text;end
(spaces must be replaced by "%20")

Trigger an application activity:
intent://#Intent;action=android.intent.action.MAIN;category=android.intent.category.LAUNCHER;launchFlags=0x10008000;package={com.name.package};component={com.name.package}/{com.name.activity};end

Example: Triggering the MediaContact communication window
intent://#Intent;action=android.intent.action.MAIN;category=android.intent.category.LAUNCHER;launchFlags=0x10008000;package=com.telelogos.mediacontact;component=com.telelogos.mediacontact/com.telelogos.mediacontactlib.McCommunicationsActivity;end

Trigger an activity application with parameters:
intent://#Intent;action=android.intent.action.MAIN;category=android.intent.category.LAUNCHER;launchFlags=0x10008000;package={com.name.package};component={com.name.package}/{com.name.activity};{type}.{parameterName}={parameterValue};end

Example: Triggering the MediaContact security window and forcing the password to 1234
intent://#Intent;action=android.intent.action.MAIN;category=android.intent.category.LAUNCHER;launchFlags=0x10008000;package=com.telelogos.mediacontact;component=com.telelogos.mediacontact/com.telelogos.mediacontactlib.McSecurityActivity;i.operation=14;S.value=1234;end

Trigger a service:
intent://#Intent;package={com.name.package};component={com.name.package}/{com.name.service};{type}.{parameterName}={parameterValue};end

Example: Starting a MediaContact communication
intent://#Intent;package=com.telelogoss.mediacontact;component=com.telelogos.mediacontact/com.telelogos.mediacontactlib.McApiService;S.CommandLine=-LC%20-C;end

To pass a parameter to an activity or service:
The syntax is Type.ParameterName with the following types:
- S.: String
- B.: Boolean
- b.: Byte
- c.: Char
- d.: Double
- f.: Float
- i.: Int
- l.: Long
- s.: Short

Each parameter is separated by a semicolon. Strings must respect the URL encoding (space replaced by "%20"). For example, to pass the string "foo moo" and the integer number "42": ;S.foo%20moo;i.42

See the Android Developer's Guide: https://developer.android.com/reference/android/content/Intent

13.3.2. CLYD commands

Launch a MediaContact process from a trigger: Only on Station
process://{process_name}

Provisioning root certificates or PKCS#12 archives (user certificates with private key)

prov_cert certificate.xml
Use this command to install root certificates from certification authorities or PKCS#12 archives including a user certificate and the private key associated with the certificate. The information is contained in the certificate.xml file (a template of this file is available in the MediaContact distribution folders). PKCS#12 certificates or archives must be sent to the device before executing this command. See the chapter "Creating a package with certificate provisioning" for more information about compatible formats, Android version constraints and certificate.xml file syntax.

Wi-Fi provisioning

prov_wifi wifi_android.xml
This command allows you to create or edit a Wi-Fi input. The information is contained in the wifi_android.xml file (a template of this file is available in the MediaContact distribution folders). Examples are given below.

The file can contain multiple <entry> items in order to configure multiple Wi-Fi networks in a single command.

prov_wifi -d ''Wi-Fi SSID''
This command allows you to delete a Wi-Fi input (except for Motorola add-on).
(For NON-Android Enterprise devices, the Wi-Fi input must have been created using the prov_wifi command. For Android Enterprise devices, it is not necessary for the WiFi input to be created using the prov_wifi command).

prov_wifi -a ''Wi-Fi SSID''
This command enables the Wi-Fi network with this SSID on the device. For the activation to work, the Wi-Fi network must have been previously configured on the device, either manually or via CLYD.

Examples of Wi-Fi configuration files:

Open network:

```
<?xml version="1.0" encoding="UTF-8"?>
<wifi>
    <entry>
        <ssid>Wi-Fi network SSID</ssid>
        <security>
            <authentication>None</authentication>
            <encryption>None</encryption>
        </security>
    </entry>
</wifi>
```

WEP:

```
<?xml version="1.0" encoding="UTF-8"?>
<wifi>
    <entry>
        <ssid>Wi-Fi network SSID</ssid>
        <security>
            <authentication>WEP</authentication>
            <encryption>None</encryption>
            <password>WEP key in hexadecimal format</password>
        </security>
    </entry>
</wifi>
```

WPA:

WPA-TKIP:

```
<?xml version="1.0" encoding="UTF-8"?>
<wifi>
    <entry>
        <ssid>Wi-Fi network SSID</ssid>
        <security>
            <authentication> WPA</authentication>
            <encryption>TKIP</encryption>
            <password>password</password>
        </security>
    </entry>
</wifi>
```

WPA-AES:

```
<?xml version="1.0" encoding="UTF-8"?>
<wifi>
    <entry>
        <ssid>Wi-Fi network SSID</ssid>
        <security>
            <authentication> WPA</authentication>
            <encryption>AES</encryption>
            <password>password</password>
        </security>
    </entry>
</wifi>
```

WPA2 Personal:

WPAWPA2 Personal-TKIP+AES:

```
<?xml version="1.0" encoding="UTF-8"?>
<wifi>
    <entry>
        <ssid>Wi-Fi network SSID</ssid>
        <security>
            <authentication>WPA2</authentication>
            <encryption>TKIP+AES</encryption>
            <password>password</password>
        </security>
    </entry>
</wifi>
```

WPA2 Personal TKIP:

```
<?xml version="1.0" encoding="UTF-8"?>
<wifi>
    <entry>
        <ssid>Wi-Fi network SSID</ssid>
        <security>
            <authentication>WPA2</authentication>
            <encryption>TKIP</encryption>
            <password>password</password>
        </security>
    </entry>
</wifi>
```

WPA2 Personal AES:

```
<?xml version="1.0" encoding="UTF-8"?>
<wifi>
    <entry>
        <ssid>Wi-Fi network SSID</ssid>
        <security>
            <authentication>WPA2</authentication>
            <encryption>AES</encryption>
            <password>password</password>
        </security>
    </entry>
</wifi>
```

WPA2 Enterprise (802.1X):

<method> must contain one of the following values: PEAP, TLS, TTLS, PWD, SIM, AKA, FAST, LEAP.

<phase2authentication> is required for the PEAP, TTLS and FAST methods. Possible values: None, MSCHAP, MSCHAPV2, PAP, GTC.

<ca_certificate> is required for the PEAP, TLS, TTLS and FAST methods.

- In Android Enterprise mode (CLYD DPC installed), this item must contain the path to the file containing the certificate from the certificate authority. This file must have been previously transferred to the device. It will be automatically provisioned by the prov_wifi command.
- In Device Admin mode (no CLYD DPC), this item must contain the alias of the certificate authority previously provisioned on the device using the prov_cert command.

<user_cert> is required for the TLS method.

- In Android Enterprise mode (CLYD DPC installed), this item must contain the path to the file containing the certificate from the certificate authority. This file must have been previously transferred to the device. It will be automatically provisioned by the prov_wifi command.

- In Device Admin mode (no CLYD DPC), this item must contain the alias of the certificate authority previously provisioned on the device using the prov_cert command.

<identity> is required for the PEAP, TLS, TTLS, PWD, FAST and LEAP methods.

<anonymous_identity> is available for the PEAP and TTLS methods.

<provisioning> is available for the FAST method.

<password> is required for the PEAP, TLS, TTLS, PWD, FAST and LEAP methods. For TLS, <password> is the password from the PCKS#12 archive (user_cert), if there is one.

WPA2 Enterprise (802.1X) EAP-TLS:

```
<?xml version="1.0" encoding="UTF-8"?>
<wifi>
    <entry>
        <ssid>Wi-Fi network SSID</ssid>
        <security>
            <authentication>EAP</authentication>
            <eap>
                <method>TLS</method>
                <phase2authentication>MSCHAPV2</phase2authentication>
                <ca_certificate> CA certificate path/alias</ca_certificate>
                <user_cert> user certificate path/alias</user_cert>
                <identity> identity (login...)</identity>
                <anonymous_identity></anonymous_identity>
                <provisioning></provisioning>
                <password>password</password>
            </eap>
        </security>
    </entry>
</wifi>
```

WPA2 Enterprise (802.1X) PEAP:

```
<?xml version="1.0" encoding="UTF-8"?>
<wifi>
    <entry>
        <ssid>Wi-Fi network SSID</ssid>
        <security>
            <authentication>EAP</authentication>
            <eap>
                <method>PEAP</method>
                <phase2authentication>MSCHAPV2</phase2authentication>
                <ca_certificate>path to CA certificate file</ca_certificate>
                <user_cert></user_cert>
                <identity> identity (login...)</identity>
                <anonymous_identity></anonymous_identity>
                <provisioning></provisioning>
                <password>password</password>
            </eap>
        </security>
    </entry>
</wifi>
```

Hidden SSID:

If the Wi-Fi network's SSID is hidden, it must be specified in a <hidden_ssid> item located at the same level as the <ssid> item.

Open network with hidden SSID:

```
<?xml version="1.0" encoding="UTF-8"?>
<wifi>
    <entry>
        <ssid>Wi-Fi network SSID</ssid>
        <hidden_ssid>True</hidden_ssid>
        <security>
            <authentication>None</authentication>
            <encryption>None</encryption>
        </security>
    </entry>
</wifi>
```

Advanced configuration: proxy and IP addresses:

By default, no proxy is configured and the DHCP server is used to assign IP addresses.
If you want to configure a proxy or set the IP address, gateway, subnet mask and DNS servers yourself, use the <advanced> item under the <security> item.

WPA2 Personal AES with advanced configuration:

```
<?xml version="1.0" encoding="UTF-8"?>
<wifi>
    <entry>
        <ssid>Wi-Fi network SSID</ssid>
        <security>
            <authentication>WPA2</authentication>
            <encryption>AES</encryption>
            <password>password</password>
        </security>
        <advanced>
            <proxy>
                <hostname>proxy hostname</hostname>
                <port>proxy port</port>
                <bypass>url 1,url 2,url 3...</bypass>
            </proxy>
            <ip_settings>
                <ip>IP address</ip>
                <gateway>gateway</gateway>
                <subnet_mask>subnet mask</subnet_mask>
                <dns1>DNS 1 IP address</dns1>
                <dns2>DNS 2 IP address</dns2>
            </ip_settings>
        </advanced>
    </entry>
</wifi>
```

Provisioning APN (Samsung devices): see19.3 page 91.

Provisioning contacts (telephone)

prov_contact -d

Delete all contacts

prov_contact file.csv

Add, modify, delete, add contact to favorites.
If the contact does not exist, it will be created. If the contact already exists (Last name, first name) it will be modified.

File format:
The file can have 4 or 7 fields, separated by semicolons.
Name;Firstname;Mobile phone;Email
or
Name;Firstname;Mobile phone;Email;Work phone;Service;Action
(Action = D to Delete the contact or F to add contact to Favorites)

Example: delete contact
Name;Firstname;Mobile phone;Email;;;D
Name;Firstname;Mobile phone;Email;Work phone;Service;D

Example: Add contact to favorites
Name;Firstname;Mobile phone;Email;;;F
Name;Firstname;Mobile phone;Email;Work phone;Service;F

Installing an Android application (application.apk)

Install <APK file path>
Requires Android Enterprise or a manufacturer signed add-on for silent installation
Example:
Install /sdcard/exitcode.apk

Uninstalling an application

Uninstall <package name>
Requires Android Enterprise or a manufacturer signed add-on for silent installation
Example:
Uninstall com.telelogos.exitcodeandroid

Provisioning .MX files (ZEBRA devices only)

prov_mx <path_file.xml>
Use this command to provision MX files received from the STAGENOW tool.

prov_mx <path_file.xml> -Attempts {number of attempts} -Delay {interval between attempts}
- number of attempts between 1 and 20, by default 1
- interval between attempts **expressed in milliseconds**: between 10 and 60000, by default 100

Example: 3 attempts at 2 second intervals
prov_mx commandes.xml -Attempts 3 -Delay 2000

Value Clyd variables in a device file

Echo %variable% > <path_file.txt>
Use this command to write Clyd variables on a file of the device.
You can use all the Clyd variables: device identifier %USER% or coordinates (%T_USERS.Name%, %T_USERS.Street%, %T_USERS.City%, %T_USERS.PostCode%, %T_USERS.Country%, %T_USERS.UserFunction%, %T_USERS.Service%, %T_USERS.Localisation%) or even %DATE%, %AliasStation=NAME_Alias%...

Example: value of the device name defined in Clyd in a "device_name" txt file
echo %T_USERS.NAME% > /sdcard/device_name.txt

Trigger a system command

Targeted devices must have a signed manufacturer add-on (Samsung, Sony and Zebra cannot be targeted with system command), otherwise system commands will not be executed on the devices.

command://{command}

Example: Triggering the CLYDMediaContact activity window
command://am start --user 0 com.telelogos.mediacontact/com.telelogos.mckiosk.McKioskInfoActivity

Example: Locking the device
command://MCLOCK

Several commands are available to execute directly on the device.

List of CLYDMediaContact commands:

MCLOCK (with or without device identifier)
Example: MCLOCK 123
Set the password found on the device information sheet

MCWIPE (with or without device identifier)
Example: MCWIPE 123
Wiping data

MCCONNECT (with or without reminder interval)
Example: MCCONNECT:10800
Connecting to the MC server
Value is optional and is used to distribute calls (value in milliseconds)

MCCLEARDATA *package_name*
Ex : MCCLEARDATA com.android.chrome
Clears application data (including history)

MCCLIENTINIT (with or without device identifier)
Example: MCCLIENTINIT 123
Initializing MC client

MCPROCESS (with the name of the process to be executed)
Example: MCPROCESS CLYD_MyProcess
Executing the process

MCINVENTORY (with or without device identifier)
Example: MCINVENTORY 123
Triggering a software and hardware inventory

MCLOCATE (without parameters)
Example: MCLOCATE
Geolocation request

MCMESSAGE (with a message)
Example: MCMESSAGE This is a message
Sending a message to the client

MCREBOOT (without parameters)
Example: MCREBOOT
Rebooting the device

MCUNLOCK (without parameters)
Example: MCUNLOCK
Removing the password protection

MCENABLEAPP (with package name)
Example: MCENABLEAPP com.exitcode.android
Activating an application

MCDISABLEAPP (with package name)
Example: MCDISABLEAPP com.exitcode.android
Deactivating an application

Chapter 14. I want to monitor my fleet (advanced)

With Clyd, you can monitor your devices, tracking usage, hardware or geolocation (“Monitoring” menu). To do this, you need to create monitoring profiles which will apply to the entire company.

14.1. Profile inheritance

When applying one or more profiles, lower levels inherit the profiles.
For example, if you apply a single company profile, all devices will inherit this profile.

If an item at a lower level already has a profile, it will not be updated.
For example, if you apply a profile at a company level, devices inherit this profile, except for those that already have a profile.

14.2. Applying the profile to the device

When applying one or more profiles, devices with a Google-ID are contacted and connect to the server to apply their profile(s). Devices without a Google-ID, or that cannot be reached via Google-ID, will apply their profile(s) when they connect to the server (according to their ***communication schedule, see 15.1 page 72***).

Profiles can be applied at the company level, in the device list, or to a specific device.

At COMPANY level
Home\Profiles menu, click on “Apply profiles”.
Select “xxx profile.” (You can select several profile types)
Select the profile from the list.
Click “Apply.”

Device list
Device menu: Select one or more devices.
Hamburger menu: Select “Apply profiles”
Select “xxx profile.” (You can select several profile types)
Select the profile from the list.
Click “Apply.”

Device details
Device menu: Click on a device to access the details.
Hamburger menu: Select “Apply profiles”
Select “xxx profile.” (You can select several profile types)
Select the profile from the list.
Click “Apply.”

14.3. Security profile

Security profile configuration via the "Monitoring" menu is only available for companies registered in Device Admin. To configure security profiles in an Android Enterprise company, see 12.2 page 51.

Define a new profile
The security profile allows you to define a number of rules on the devices.

Settings: Password management
No modification: No configuration of password management. If a password or a password rule has been previously configured on the device, this "No modification" choice will have no effect on the current configuration.
Remove Device Lock: Remove any lock patterns and rules configured on the device.
Force password: Enter and confirm the devices password. Once validated, the password is no longer visible.
Specify rules: Allows you to define rules for device passwords. The password will then be chosen by the owner of the dvice according to the defined rules. There are two types of rules:

- Password quality: valid on all versions of Android, password quality consists of specifying rules concerning the type of characters imposed and the minimum length of the password.

- Password complexity: valid from Android 12, password complexity overrides password quality settings. Password complexity consists of choosing between three levels of security:
    - Low complexity: it allows at least:
        - Reason for unlocking;
        - PIN code that can contain repetitions (4444…) or ordered sequences (4321, 2468…).
    - Medium complexity: it allows at least:
        - PIN code of at least 4 characters that cannot contain repetitions (4444…) or ordered sequences (4321, 2468…);
        - Alphabetical password of at least 4 characters;
        - Alphanumeric password of at least 4 characters.
    - High complexity: it allows at least:
        - PIN code of at least 8 characters that cannot contain repetitions (4444…) or ordered sequences (4321, 2468…);
        - Alphabetical password of at least 6 characters;
        - Alphanumeric password of at least 6 characters.

Settings: Detect rooted devices
The CLYDMediaContact client checks whether the device has been rooted. You can define an action to be taken if the device has been rooted.
None: No action.
Reset to factory values (wipe): Automatic device reset.
Lock the device by password change: Force a password on the device.

Settings – USB security: (only for Samsung devices)
Enable USB port blocking: Block file transfers via USB. The battery can still be charged.

Settings – System
Block system updates: Block system updates from Samsung.
Block safe mode: Block "Safe Mode" (keyboard combination which resets the tablet to factory settings) (Android Enterprise/Samsung)
Block debug mode: Prohibit activating the Debug mode. (Android Enterprise)
Prohibit unknown sources: Prohibit activating unknown sources. (Android Enterprise)

Firewall – List of authorized domains:
All domains are authorized by default.

Firewall – List of forbidden URLs:
All URLs are authorized by default.

Buttons: (only for Samsung devices)
Blocks access to certain keys on the device

System
Block system updates: Blocks updates from Samsung (Samsung)
Block Safe Mode: Prohibits activation of Safe Mode (key combination used to reset the tablet to Factory Mode). (Samsung)

CLYDMediaContact Client
Require a password for accessing options: Denies access to the CLYDMediaContact client administrator functions (reset, call settings, configuration)
Make CLYDMediaContact client notifications non-clickable: Denies user to access to the CLYDMediaContact app by clicking on the notification.

View profile name

Device list
Device menu.
Add the "Security Profile" column to view the profile name of each device.

Device details
Device menu: Click on a device to access the details.
The profile name is displayed on the device drawing, by hovering over the monitoring profile icon.

14.4. Monitoring profile

❖ **Define a new profile**

The monitoring profile is used to monitor device events or components. For events, the date and time are recorded. For component monitoring, a value is periodically recorded. The information collected is fed back to the CLYD server each time the device is connected (see 15.1).

General:

By default, the monitoring profile is always valid. The validity period allows you to define shorter monitoring periods (for example: Enable monitoring between 08:00 and 18:00, Monday to Friday only).

Monitoring log conservation allows you to define a retention period for the monitoring information collected (1 to 99 days).
This relates to the retention of monitoring data on the device, where the device cannot connect to the CLYD server. Otherwise, the data is fed back each time the device is connected.
On the server side, the monitoring data is retained for 30 days.

Events monitoring:

Device (stopping, starting)
CLYDMediaContact Client (stopping, starting)
Standby (Standby/Exit)
WiFi (connection/disconnection)
Cellular data (connection/disconnection)

Periodic monitoring:

A period is defined for each monitored item.
Battery
Free memory
CPU load
WiFi level
Cellular data level

Use "battery" monitoring to display a message on the device according to a threshold.

Usage monitoring:

Usage time / application
Mobile data usage / application
Global mobile data usage

Caution:

- Usage monitoring data is collected once an hour on your devices.
- Monitoring data on usage associated with web applications cannot be distinguished from other usage on your web browser. You can view usage times and mobile data for your web applications by monitoring the web browser application used.

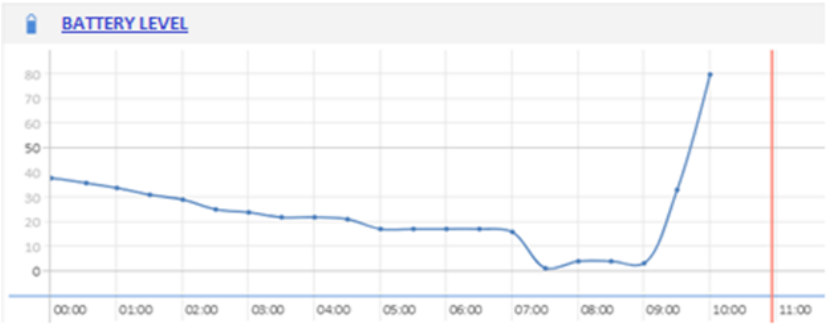
❖ **View profile name and monitoring data:**

Events and periodic monitoring

The monitoring information can be consulted on the Device details in the "Monitoring data" tile. You can select a time period. Data from the last 24 hours are displayed by default.

Example of monitored data: Battery

Usage monitoring:



This type of monitoring makes it possible to trace the daily durations and data use to the servers. Unlike other types of monitoring, this data is not viewable on the device detail but using the snaps-in from the Clyd dashboard or homepage.

14.5. Geolocation profile

To geolocate your business terminals, make sure you comply with the regulatory framework in which you operate (GDPR in Europe). In particular, you need to inform your employees about this tracking and the options they have for deactivating it (this is always possible from the "Information" menu in the MediaContact client).

Define a new profile

The geolocation profile allows the geolocation coordinates to be fed back in order to locate the device on a map. CLYD stores geolocation data that can be used to display a device's route.

General:

By default, the geolocation profile is always valid, the validity period allows you to define shorter geolocation periods (for example: Enable geolocation between 08:00 and 18:00, Monday to Friday only).

Obtain device position allows you to define an interval within which to obtain the device coordinates. Data is sent back to the server each time a call is made, or each time the terminal file is opened, if a geolocation profile is valid at the time of opening. "Ignore a position" allows you to ignore coordinates that are too close.

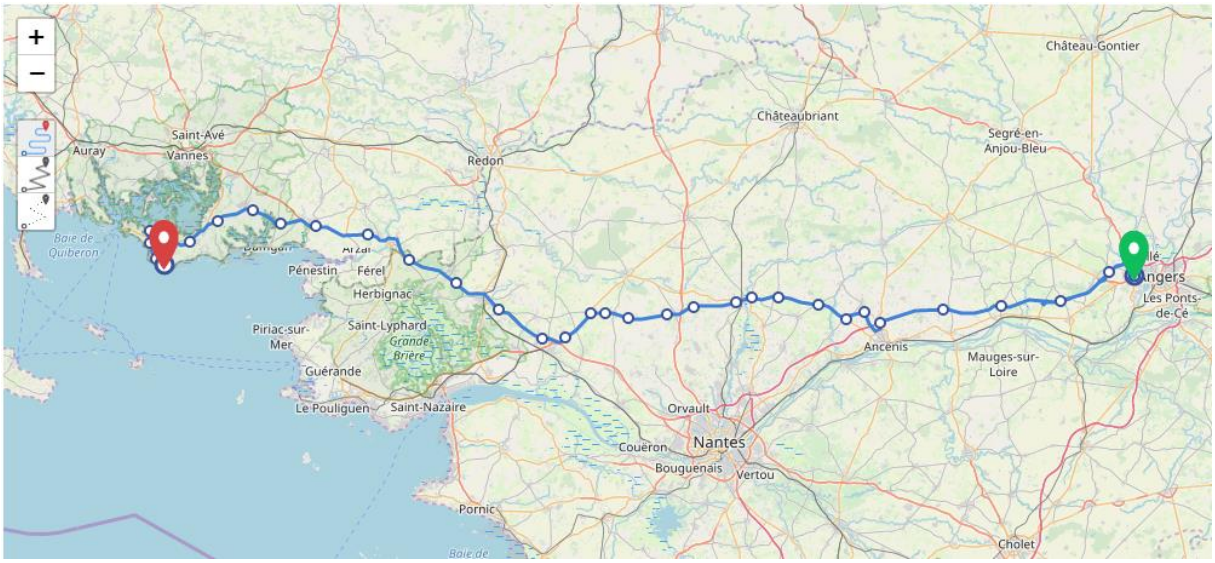
Retention period: allows you to define a retention period for the coordinates (1 to 99 days).
This relates to the retention of geolocation data on the device, where the device cannot connect to the CLYD server. Otherwise, the data is fed back each time the device is connected.
On the server side, the geolocation data is retained for 30 days.

Allow the device user to disable geolocation: the device user can interrupt the geolocation.

View profile name and geolocation information.

The geolocation information can be consulted on the Device details in the "Geolocation" tile.

Example of a device route:



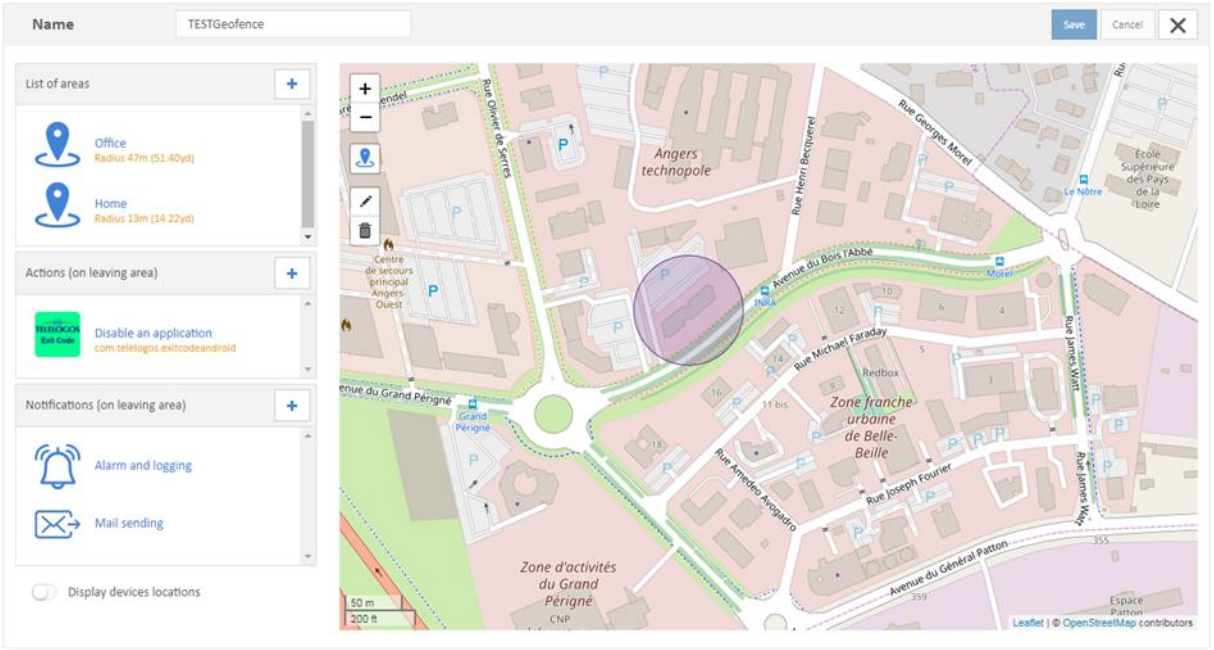
By default, a route is shown as a straight line. If you want to follow the exact route (avancé – sur un serveur dédié):

Create an account on: https://graphhopper.com/dashboard/#/register
Create a key in the "API Keys" menu
Copy and paste the key into the "ProviderRoutingGraphHopperAPIKey" field of the appSettings.config file (Path appSettings.config: C:\Program Files (x86)\Telelogos\CLYD)
Reboot IIS

14.6. Geofencing profile

The geofencing profile defines an area or areas where the device is supposed to be located. When the device leaves one of the areas, you can launch actions on the device (wipe, lock, etc.) and be informed via notifications.

Once it detects an exit from the area, the event is sent to the CLYD server in real time via Web services. If the Web services are not active, the event will be sent to the server the next time the device communicates.



List of areas: Create areas either by drawing on the map or by providing the coordinates and radius.

Actions: List of actions that can be performed on the device when leaving the area.

Lock the device (See the unlocking code on the device information sheet).
Reset the device.
Wipe an application's data.
Disable an application.
Uninstall an application.
Execute a command.

To unlock the device or reactivate an application, go to the device Action menu.

Notifications: To notify the administrator.

Alarm and logging: This is active by default. It will send you a notification via the CLYD interface and update the device log.

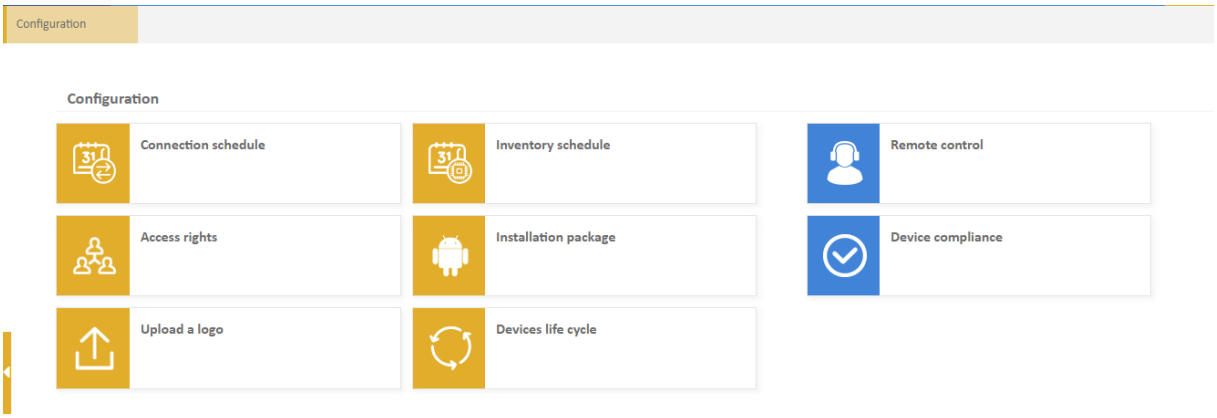
Mail sending: To notify an administrator by email (requires the configuration of an SMTP server, accessible from CLYD's Company menu.)

The profile name and geofencing status for this device are visible on the Geofencing tile of the device information. The possible statuses are: Activation in progress/In area/Out of area/Never entered the area/Unable to activate.

Chapter 15. I want to configure additional parameters

The configuration menu gives access to 7 different tabs. The functionalities of each of these tabs are detailed in the corresponding sub-chapters below.

- "Connection schedule" tab (advanced): see 15.1 after.
- "Inventory schedule" tab (advanced): see 15.2 after.
- "Access rights" tab (advanced): see 15.3 after.
- "Installation package" tab: see 4.2 page 15.
- "Upload logo" tab: see 15.4 after.
- "Device lifecycle" tab (advanced): see 15.5 after.
- "Remote control" tab (advanced): see Chapter 17 page 98.
- "Device compliance" tab: see 15.6 page 78 after.



15.1. I want to define a specific connection schedule between my devices and the server (advanced)

The connection schedule allows your devices to connect regularly to the CLYD server.

15.1.1. Connection schedule definition

Uses for the connection schedule:

Devices with Google-ID:

- ❖ Deploying a kiosk, a WorkSpace or a provisioning profile and applying profiles require the use of a Google-ID. If the device cannot be woken by this means, connection via the connection schedule will be used.
- ❖ Regular update of "Status" information (Battery level, Disk space, etc.) of the device information sheet.
- ❖ Applying monitoring profiles on the device.

Devices without Google-ID:

- ❖ Deploying a kiosk, a WorkSpace or a provisioning profile.
- ❖ Monitoring profile application.
- ❖ Regular update of "Status" information (Battery level, Disk space, etc.) of the device information sheet.

Regularly connecting your devices also allows for better monitoring of your device population. The plug-in components allow you to view how often your devices connect (Connection status for the current day, Status of last connections, etc.).

The connection schedule is associated with a target (set of devices). It is possible to configure several settings:

Call frequency
The frequency can be expressed in Minutes, Hours or Days.

Call attempts
Used when connecting to the CLYD server in case of failure.

Call distribution
An algorithm is used to distribute the connection of all the target's devices to avoid an excessive load on the CLYD server. (Without this setting, all the target's devices will connect to the CLYD server at the same time.)

Validity periods
Allows you to reduce connections for specific time periods or days of the week.

Default device connection: a default connection schedule is available whereby all of your devices connect to the CLYD server every 4 hours between 06:00 and 22:00 with 3 hours call distribution.

Restrictions imposed by the global administrator for company administrators:

15.1.2. I want to set connection schedules rules for the companies on my server (advanced)

To set rules for your users' connection schedules, you need to be logged in as a global administrator.

You can then :

- Limit call frequency to less than one hour. This limits the impact of device connections on your server's activity.
- Force call distribution to three-quarters of the values entered. This allows you to distribute your devices' connections to your server over a period of time.

15.2. I want to set up a hardware inventory schedule on my devices (advanced)

Inventory planning lets you schedule a complete hardware inventory survey on your devices.

The inventory schedule will be associated with a target (set of devices), and you can define several parameters:

Inventory frequency
Frequency can be expressed in Days or Weeks.

Validity ranges
Allows you to select the days of the week on which the inventory will be taken.

If the device is not started at the time of the inventory, the inventory is taken within 23 hours if the frequency has been set in days, and within 6 days if the frequency has been set in weeks.
No communication between device and server is triggered during scheduling. Inventory data will be uploaded to Clyd at the next scheduled or manually triggered communication.

Default device connection: a connection schedule is available by default, and a hardware inventory schedule is drawn up every Monday at 12 noon.

15.3. I want to set Clyd access rights (advanced)

The access rights tab of the Configuration menu lets you create Clyd accounts, modify their rights or set passwords for them.

15.3.1. Authentication type

CLYD supports three authentication types:

- MediaContact (authentication against the MediaContact database).
- Windows (authentication against an Active Directory or a local SAM database).

- LDAP (authentication against an LDAP directory).

During a new installation, CLYD uses MediaContact's integrated authentication and creates an "admin" account (password: admin) with administrator rights.

In the case of a CLYD installation on a running MediaContact database, the authentication type is not changed, but only the MediaContact Global administrator is able to connect to the CLYD console. The access rights to the CLYD console must be created in the CLYD console.

The authentication type can only be changed from the MediaContact console.

15.3.2. Type of access rights

CLYD allows you to grant 3 types of access rights to the accounts:

- Administrator (Create, Modify, Read, Execute, Remote Control)
- Carrier (Read, Execute, Remote Control)
- Remote control user (Read, Remote Control)
- User (Read)

Administrators have full access rights to Companies, Devices, Targets, Dashboard, Catalog, Kiosk and WorkSpace, Monitoring, Provisioning and all configuration elements (Connection schedule, Logo, Generating an installation package, Access rights, Remote control and Device compliance).

In addition to Read rights, operators have kiosk, WorkSpace and monitoring profile deployment rights, monitoring profile application rights, Modification of company and device details and Device execution rights (Delete, Unassign, Lock, Reset/Wipe, Inventory Request).

Users only have Read rights.

Please note that only the "Devices", "Dashboard" and "History" menus are visible to users with view rights. An operator or user on view is therefore unable to access the kiosks.

15.3.3. Level of assignment of access rights

Access rights are granted at different levels:

- GLOBAL level : GLOBAL level provides access to all COMPANIES and VIEWS. It gives access to all functions, depending on the type of access rights granted (Administrator, Operator or User).
- COMPANY level : COMPANY level provides access to a specific COMPANY and its VIEWS. It gives access to all functions, depending on the type of access rights granted (Administrator, Operator or User).
- VIEW LEVEL : VIEW level provides access to all devices in the VIEWS and/or at the VIEW levels. It grants EXECUTE access only to devices, depending on the type of access rights granted (Operator or User). Administrator rights cannot be granted at VIEW level.

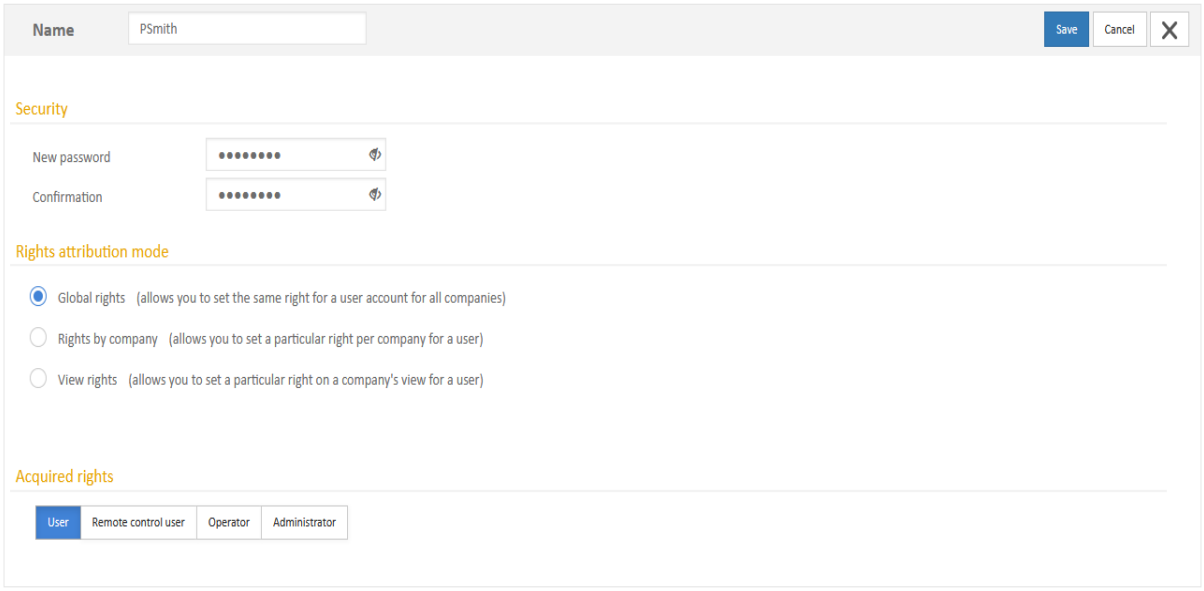
The Admin account (created automatically during CLYD installation) allows you to define access rights at GLOBAL, COMPANY and VIEW levels.

15.3.4. Rights at GLOBAL level

Prerequisite: be logged in as a GLOBAL administrator.

Select GLOBAL level from the drop-down menu:

In the Configuration menu – Access rights:



Name: Give an account name. In the case of Windows authentication, you can browse the directory and select an existing account or group (not supported for LDAP accounts).

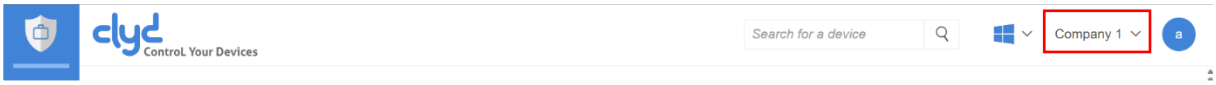
Rights allocation mode: Global rights

Acquired rights: Select the type of access rights required (Administrator, Operator or User)

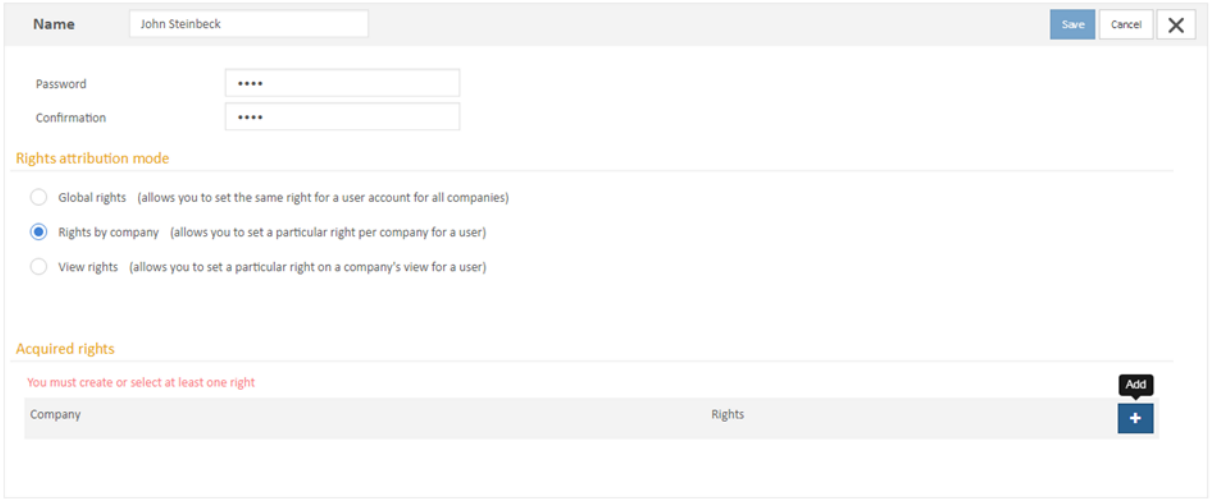
15.3.5. Rights at COMPANY level

Prerequisite: be logged in as an administrator.

Select the COMPANY level from the drop-down menu.
As a GLOBAL administrator, you have the same options by keeping “Global Level” in the drop-down menu.



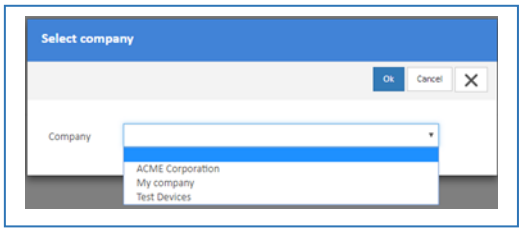
In the Configuration menu – Access rights:



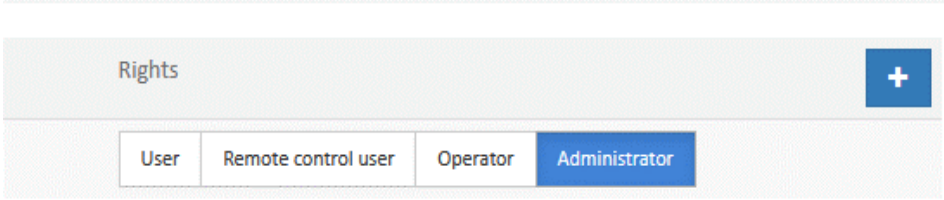
Name: Give an account name. In the case of Windows authentication, you can browse the directory and select an existing account or group (not supported for LDAP accounts).

Rights allocation mode: Access rights by company

Acquired rights: Select a company by clicking on +



Then select the desired access rights.



The account and its associated access rights will only be visible in the company access rights tile. They will not be visible at Global level.

15.3.6. Multi-company Rights

Prerequisite: be logged in as a global administrator or a multi-company administrator.

As a GLOBAL administrator, CLYD allows you to grant access rights to multiple companies.

Giving access rights to multiple companies.

Select GLOBAL level from the drop-down menu.
In the Configuration menu – Access rights
Rights allocation mode: select "Rights by company"
Acquired rights: add a company, select the type of access rights.

You can add other companies with potentially different types of access rights. An administrator of company No. 1 can also be the Operator of company No. 2.

Adding rights to an account that already has rights to one or more companies.

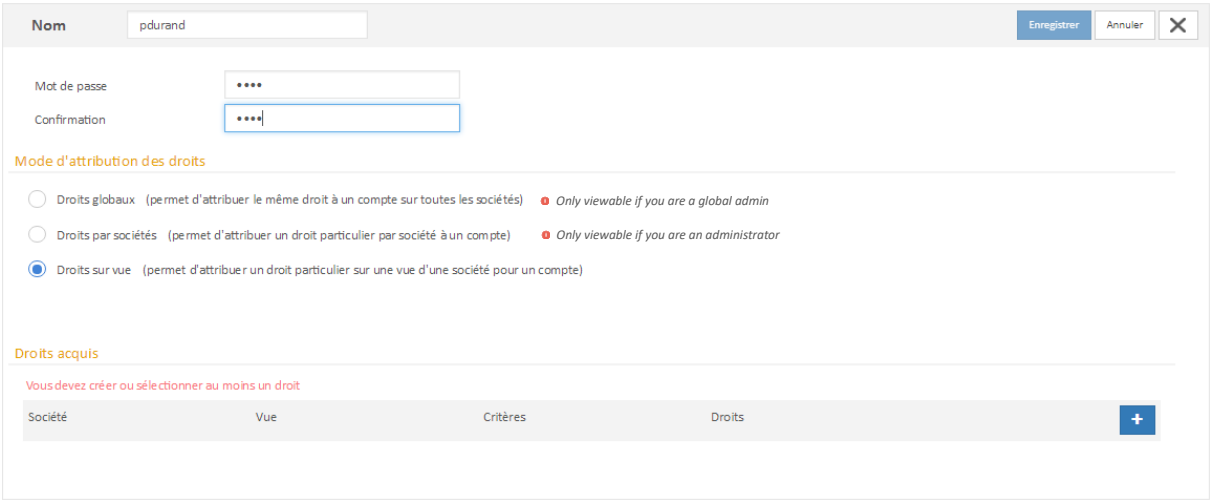
When you give access rights to an account on a company, the account is only visible in the company where it has those rights.
To add access rights to an existing account, you must go to the company where the account has those rights (Select the company from the drop-down menu), then edit this account to add a company and the required type of access rights.

15.3.7. Rights at VIEW level

Prerequisite: be logged in as an administrator.

Select the COMPANY level from the drop-down menu.
A VIEW must be configured for your COMPANY
As a GLOBAL administrator, you have the same options by keeping "Global Level" in the drop-down menu.

In the Configuration menu – Access rights:

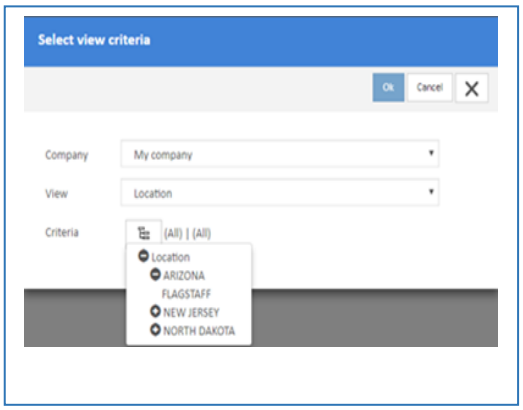


Name: Give an account name. In the case of Windows authentication, you can browse the directory and select an existing account or group (not supported for LDAP accounts).

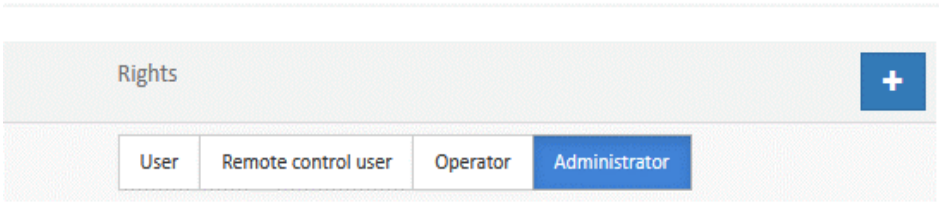
Rights allocation mode: VIEW rights

Acquired rights: Select the view level by clicking on +

Select the Company.
Select the View.
Select the View level (criteria).



Then select the desired access rights.



You can give rights on several views in the same non-global account. In this case, the right must be the same for all views.

15.4. I want to customize my company display by uploading a logo

The "Upload logo" tab in the Configuration menu lets you customize the Clyd interface for your company.

The uploaded file becomes the company logo and replaces the name of your company in the drop-down menu.

You can replace the name of your company with a logo in the CLYD application banner.

This logo must be in PNG format and have a 46-pixel height.

15.5. I want to manage the lifecycle of my devices (advanced)

The life cycle function allows you to assign a status to a device and to trigger actions when a device reach one status. By default, the device has no status.

Tab [Configuration] [Life cycle]
You can configure the system to record the history of status changes (Default: 12 days).
You can associate an action with a status (default: no action is associated with a status).

Possible statuses: Production, Stock, Maintenance, Lost, Irreparable.
Possible actions: None, Deactivate a device, Unassign a device, Activate.

The devices are activated by default.

Unassign a device: The device is no longer managed by CLYD. Its license is therefore released. For the device to be managed again by CLYD, it must be reactivated on the server and the CLYDMediaContact client must be reset on the device.

Deactivating a device: The device is still managed by CLYD, but it is no longer affected by deployments. It must be reset on the server. No action required on the device.

Configuration example:
Status: Production/Action: Activate
Status: Maintenance/Action: Deactivate the device
Status: Lost/Action: Unassign the device

Modifying the status of a device:

On the "Life Cycle" tile of the device file, the status of the device can be modified and the associated action is displayed.
You can add a comment and display files (after-sales service feedback form, etc.)

A plug-in component on the home page allows you to view the different statuses of your devices

15.6. I want to define compliance criteria on my fleet

15.6.1. Select the compliance criteria.

You can define compliance criteria to quickly identify whether the devices are operational. This information will be visible in the device lists as well as in the device details. A green mark indicates that the device is compliant, a red mark indicates that the device is non-compliant.

The set of criteria must be met for the device to be compliant.

Available criteria:

Kiosk activated/deactivated
Kiosk activated : The device is secure.
Devices with a kiosk whose status is activated are compliant.

Kiosk deactivated : The device is no longer secure.
Devices without a kiosk and devices with a kiosk whose status is deactivated are compliant.

Kiosk version
Check the kiosk version. The device is compliant if the version matches.

WorkSpace
Workspace distributed: devices with WorkSpaces will be compliant.

WorkSpace version
The WorkSpace version is checked. The terminal is compliant if the version matches.

Security profile
Security profile enabled: The terminal is secure, a security profile is deployed and enabled.

Security profile deactivated: The terminal is no longer secure.
Terminals without a security profile, or whose status is paused, will be compliant.

Security profile version
The security profile version is checked. The terminal is compliant if the version matches.

CLYDMediaContact client version
Check the CLYDMediaContact client version. The device is compliant if the version matches.

Date of last connection
The date and time of each communication by the device (date of last communication) are recorded.
The device is compliant if the connection was successful before the indicated limit (in hours or days).

App version
The version of one or more applications available on the device is checked compared to known versions on the Clyd catalog.
The device is compliant if the version matches.

If the application cannot be found on the device, then the status is non-compliant.

15.6.2. Checking the compliance of my devices

The compliance check can be performed in the company device list or in the device details.
The device list contains a column indicating whether or not the device is compliant. If the device is compliant, a green mark is displayed. If the device is non-compliant, a red mark is displayed. This information is also displayed in the device details with the same graphic presentation.
If the device is non-compliant, hovering over the red mark will bring up a tooltip containing the invalid criteria.

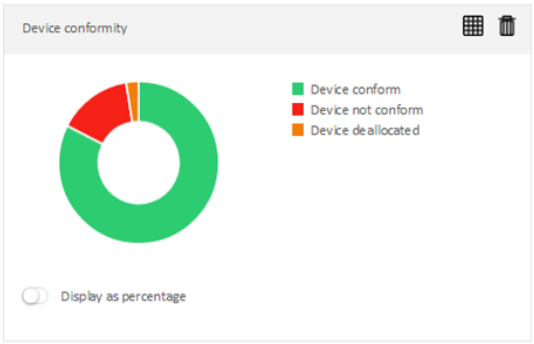
Only operational devices have a compliant or non-compliant status. Unassigned devices will be assigned a gray mark.

The plug-in component "Device compliance" allows you to view the compliance of all your devices.

Chapter 16. I want to display the status of my fleet

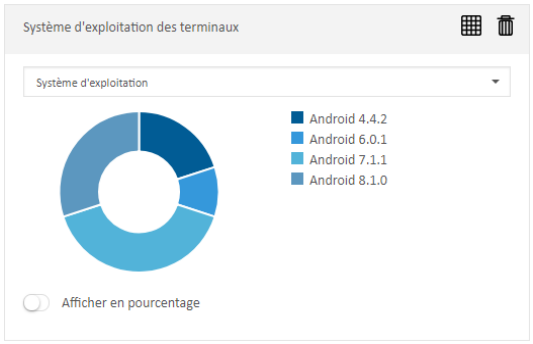
Plug-in components can be displayed by the global administrator and by the company administrator, on the home page (maximum 3 components) and on the Dashboard tile. Operators and users will be able to view these plug-in components.
There is no limit to the number of components on the dashboard; you can add more by clicking on the "+". A cogwheel lets you choose the data refresh rate.
On both the home page and the dashboard, you can:

- Click on an area of a graph to retrieve details of the devices concerned and export the list.
- Click on the grid to retrieve details of all relevant devices in a list view.
- Delete a component by clicking on the recycle garbage can



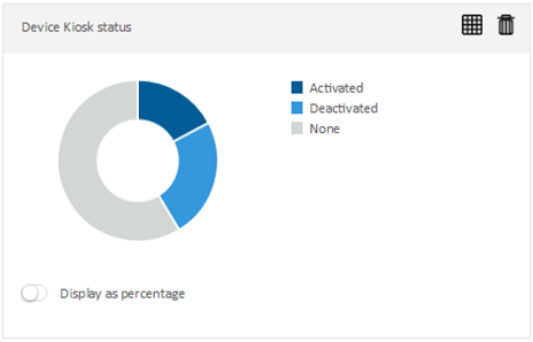
Device compliance
This plug-in component provides a graph (donut chart) showing the status of my device population, according to the compliance criteria validated by the administrator. Three possible values:

Compliant : Devices match the compliance criteria
Not-compliant : Devices do not match the compliance criteria
Unassigned : Deactivated devices



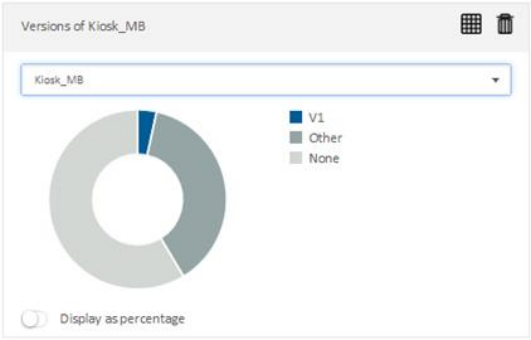
Device properties
This plug-in component provides a graph (donut chart) showing the different properties of a device (coordinates, brand, model and operating system)

Select the component and then the desired property (Example: operating system)



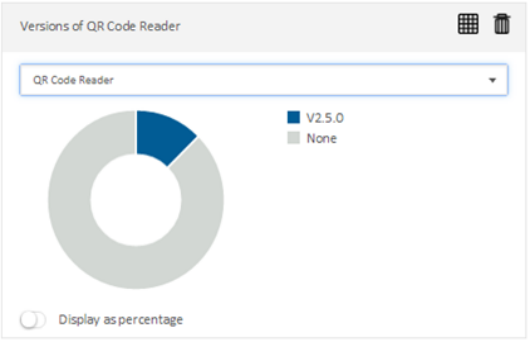
Device kiosk status (Activated/Deactivated/None)
This plug-in component provides a graph (donut chart) showing the status of the kiosk on my device population. Three possible statuses:

Activated : The kiosk is operational on the device, the device is secure.
Deactivated : The kiosk is no longer operational on the device. The device is no longer secure.
None : No kiosks have been deployed on the device.



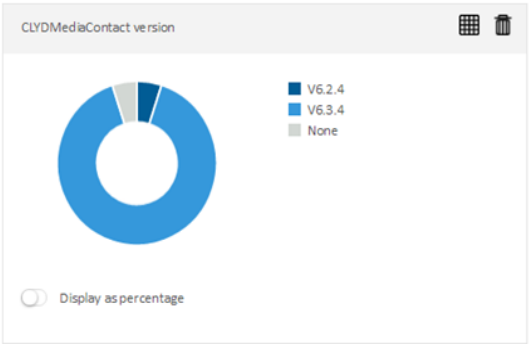
Kiosk, WorkSpace or EMM Profile version
This plug-in component provides a graph (donut chart) showing the different versions of a kiosk, WorkSpace or EMM Profile deployed on my device population. Several possible values:

Vx : Kiosk version
Other : Devices with another kiosk
None : Devices with no kiosk

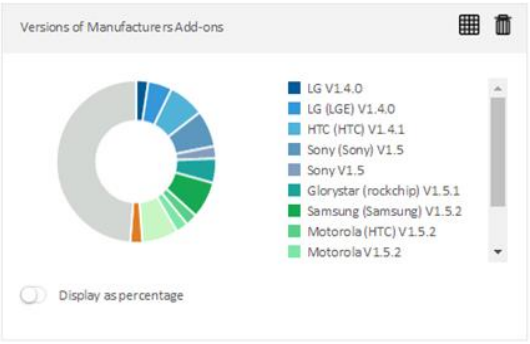


Application version (Non-system apps)
This plug-in component provides a graph (donut chart) showing the different versions of a non-system application on my device population. Several possible values:

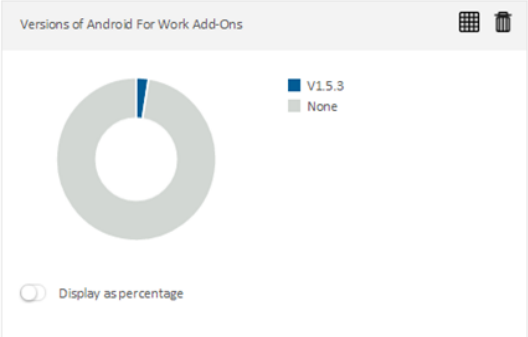
Vx : App version
None : Devices that do not have this application



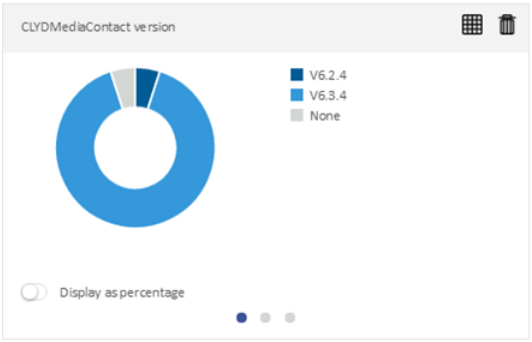
CLYDMediaContact version
This plug-in component provides a graph (donut chart) showing the different versions of the CLYDMediaContact client.



Manufacturer's add-on versions
This plug-in component provides a graph (donut chart) showing the different versions of Manufacturer add-ons.

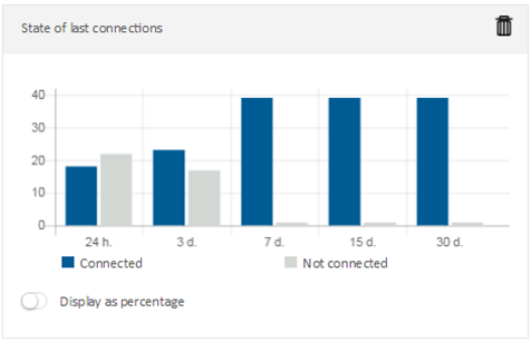


Android for Work add-on versions
This plug-in component provides a graph (donut chart) showing the different versions of the Android For Work add-on.

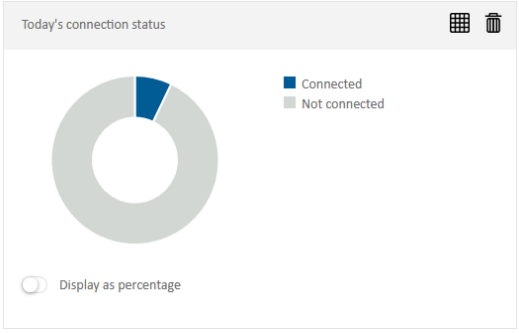


Telelogos app versions
This plug-in component includes plug-ins from CLYDMediaContact client versions, manufacturer add-on versions, and Android for Work add-ons in carousel mode.

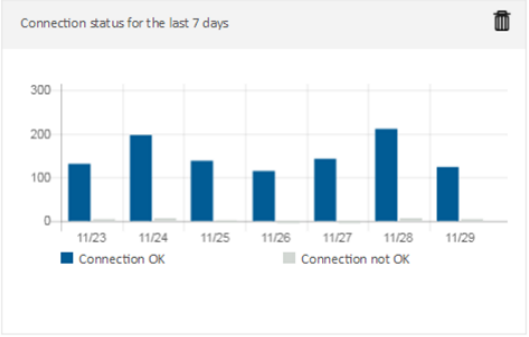
The three dots allow you to move from one component to another.



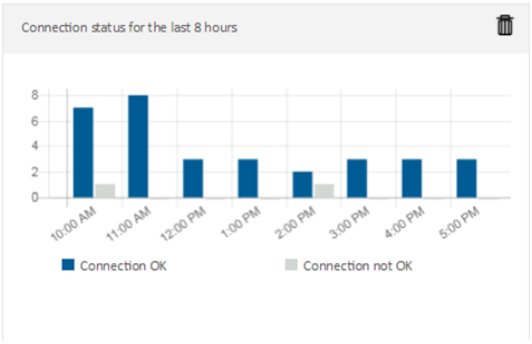
Status of last connections (24 hours - 3, 7, 15 and 30 days)
This plug-in component shows the number of devices connected or not connected during periods of 24 hours, 3, 7, 15 and 30 days.



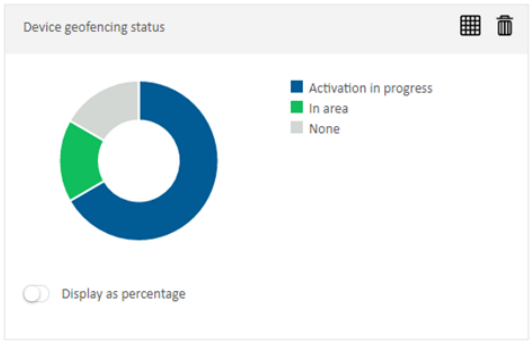
Today's connection status
This plug-in component allows shows the devices that have connected at least once that day (from midnight up to the current time).



Status of connections for the last 7 days
This plug-in component shows the total number of OK and Not OK connections, over the last 7 days, in 1-day increments.

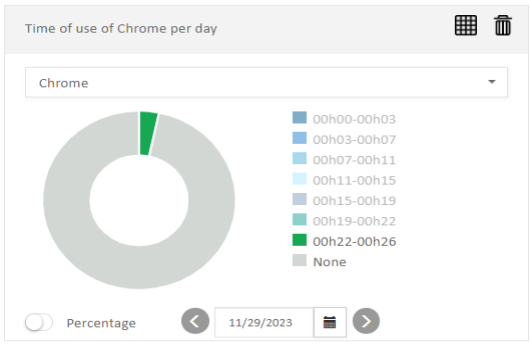


Status of connections for the last 8 hours
This plug-in component shows the total number of OK and Not OK connections, over the last 8 hours, in 1-hour increments.



GEOFENCING: Status of the devices
Use this plug-in component to view the Geofencing status of the devices.

Activation in progress : The profile application request is being processed
In area : The device is within its reference area
Out of area : The device has left its reference area
Unable to activate : Error when activating the device
None : No associated profile



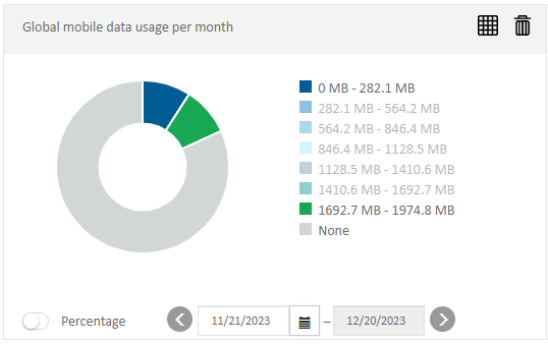
App usage time per day
This snap-in allows you to view the duration of use of an application over a day.

The graph almost always shows 8 sectors.
- The devices of sectors 1 to 7 are calculated according to the shortest time of use (excluding zero value) and the longest time of use among the company's devices for the selected day and for the selected application. Each sector will cover approximately 1/7th of this time interval.
- The 8th sector, gray in color, corresponds to devices for which the application usage time is zero.

Example :
The smallest time of use being equal to 0h04 and the largest being 1h10 (i.e. an interval of 66 minutes) for the "Forest" application on 04/05/2023, a sector will cover 1/7th of this interval or approximately 9.43 minutes or 9 minutes and 26 seconds.

All of the company's affected devices are taken into account, even those for which no usage data is present in the database for the selected application and the selected day (the latter therefore have a duration of use equal to 0).

Global mobile data usage by month



This snap-in allows you to visualize all the mobile data usage for your fleet over a one-month sliding period of your choice.

The graph almost always shows 8 sectors.
- The boundaries of sectors 1 to 7 are calculated between zero data usage and the highest usage among the company's devices for the selected day and for the selected application. Each sector will cover approximately 1/7th of this time interval.
- The 8th sector, grey in color, corresponds to devices with zero application usage.

Data is displayed in MB, rounded to the nearest 0.1 MB. You can find the exact usage in KB in the component details.

All the company's affected devices are taken into account, even those for which no usage data is present in the database for the selected application and the selected day (these therefore have a usage time equal to 0).

Global mobile data usage per day

This snap-in behaves strictly identically as the *Mobile data usage per month* component but allows you to target a specific day.

Mobile application data usage per month
Application mobile data usage per day

These two snap-in components are identical to the *Application Mobile Data Usage by Month* and *Application Mobile Data Usage by Day components*, but they allow you to monitor mobile data usage for a particular application.

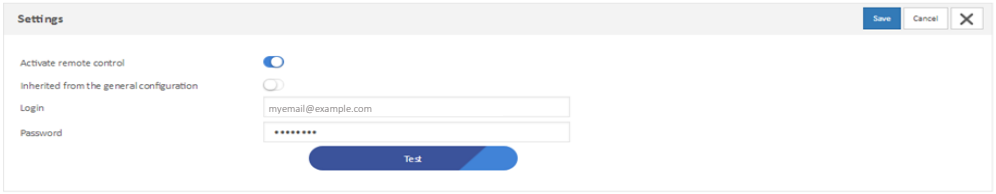
Chapter 17. Installing and configuring WiseMo with Clyd

i Installing and configuring Wisemo on devices is a prerequisite for remote control of devices. Before configuring it on devices, you must also activate remote control and configure your account at company level.

17.1. I want to configure the remote-control on my company

In the Configuration menu, "Remote control" tab, you can enable remote control for your company and set the parameters for connection to MyCloud, the server that manages remote control.

Activate/deactivate remote control: Need to be activated to use remote control from you devices (MyCloud or LAN) (see 9.2 page 42).
Login/Password: enter your information (MyCloud only). You can test the connection to validate the settings.



MyCloud licenses:

In myCloud mode, when the WiseMo Host application connects for the first time to WiseMo's myCloud servers, a myCloud license is automatically assigned to this device.

When a device associated with a myCloud license is deleted, reset to factory values or decommissioned from the actions menu of Clyd or via the actions related to the life cycle, the myCloud license associated with this device is then automatically released which will allow the myCloud server to assign it later to another device.

17.2. I want to configure Wisemo on my devices using the WiseMo Host managed configuration (recommended method for Android Enterprise)

Prérequisite : Using Wisemo 20.x

In Android Enterprise mode, the easiest way to install and configure WiseMo Host is to use the managed configuration (see 11.3.2) of the WiseMo Host application found on the Play Store. See the "Managed configurations" chapter for more information on this topic.

After adding the "WiseMo Host" application to your enterprise store, you need to edit the managed configuration of this application and fill in the following information:

Configuration name: *indicate the name you want to give to this configuration*

Guest Access Authentication Method: "Shared password"

Password: *password, or empty if no password*

Confirm access: *deactivated, so that you do not need to confirm access*

myCloud profile:

myCloud Service URL: *information available in the "Settings → Connection" menu of your myCloud space*

Account name: *information available in the "Settings → Connection" menu of your myCloud space*

Domain name: *information available in the "Settings → Connection" menu of your myCloud space*

Password: *information available in the "Settings → Connection" menu of your myCloud space*

Host name:

Naming mode: "Enter name"

Enter name: %COMPANY%-%USER%

Host licensing:

License mode:

"myCloud licensed" *if you do not use the perpetual license key*

"License key" *if you are using a perpetual license key (also gives access to myCloud if the settings in the "myCloud profile" section have been filled in). Perpetual licence key are necessary for Hosts connected in LAN mode only (without internet access)*

License key: *perpetual license key if you are in "License key" mode*

Stop Host if no activity: 0 (zero)

Automatic wake up : *Enable to aceppt any connection on the Host without the Host being started. The option is disable by default.*

Maintenance password:

Password: *password useful to access the Host parameters.*

Protect configuration changes *Disable. By default, the option is enabled and force to use the password set to access to the Host parameters.*

Protect Host service: *Enable. Protect Host options "Host Start, Pause, Restart, Exit and Disconnect". Disable by default.*.

Service Notification is clickable: Disable. By default, this option is enabled and allows you to be redirected to the Wisemo agent interface on notification click.

You need to deploy the Wisemo Host apk on your devices and an addon corresponding to the brand of your device via a WorkSpace.

If you need to deploy WiseMo on different brands of devices, we recommend creating as many WorkSpaces or EMM profiles as brands, associating them with brand-based targets.

17.3. I want to configure Wisemo for devices in closed environment (using the host.xml file)

WiseMo can be an integral part of the CLYD installation package. This allows it to be installed and configured automatically during the deployment of the CLYDMediaContact client.

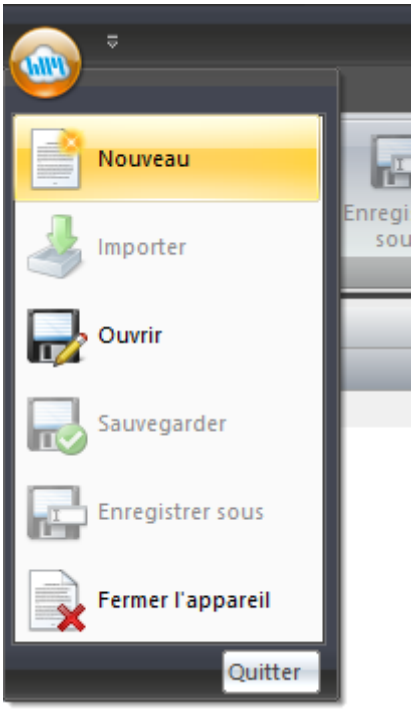
To do this, you must:

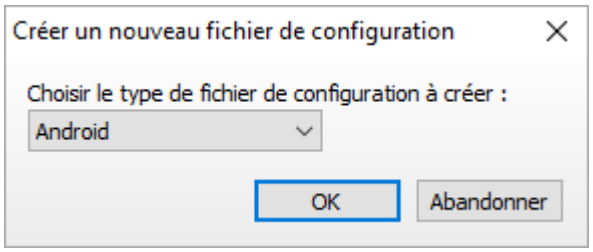
- Prepare a host.xml file (configuration file for the WiseMo client)
- Add this file and the WiseMo APK to the CLYD installation package.

Preparing the host.xml file

WiseMo provides the "Mobile Host Manager" tool used to create the host.xml file.

1 – Create a new host.xml file by selecting the Android platform.

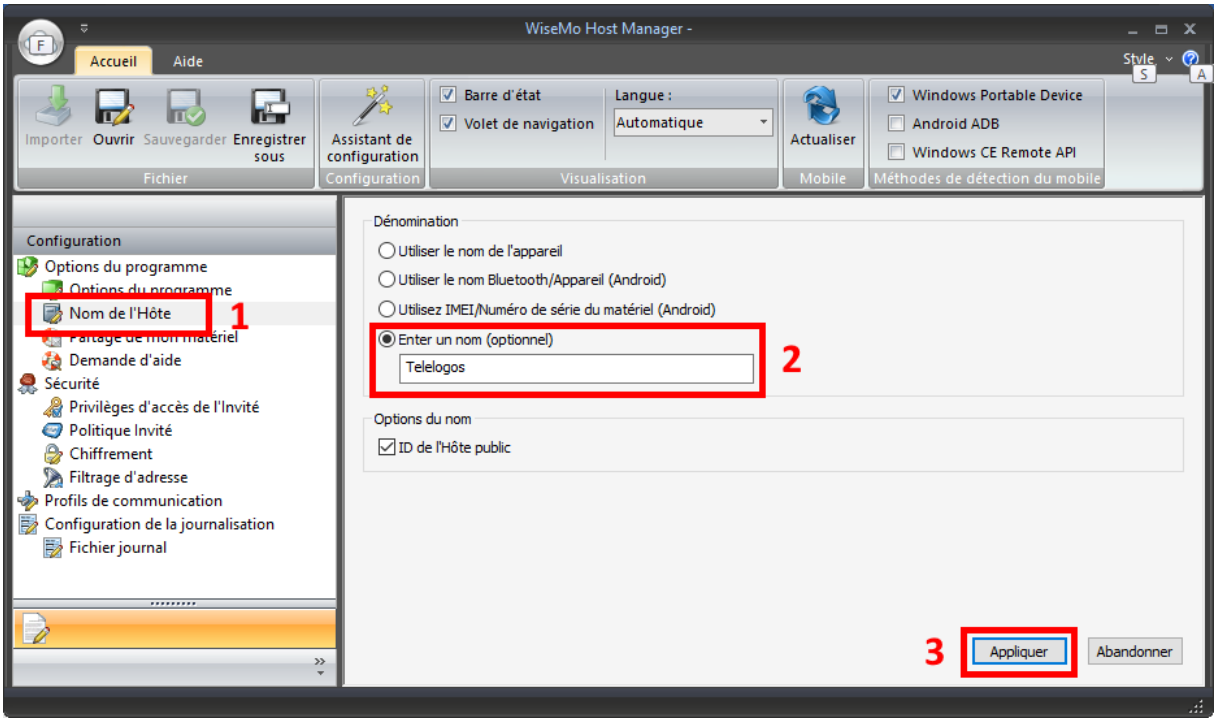




2 – Use the configuration wizard to customize the host.xml file according to your needs (password, license, etc.).

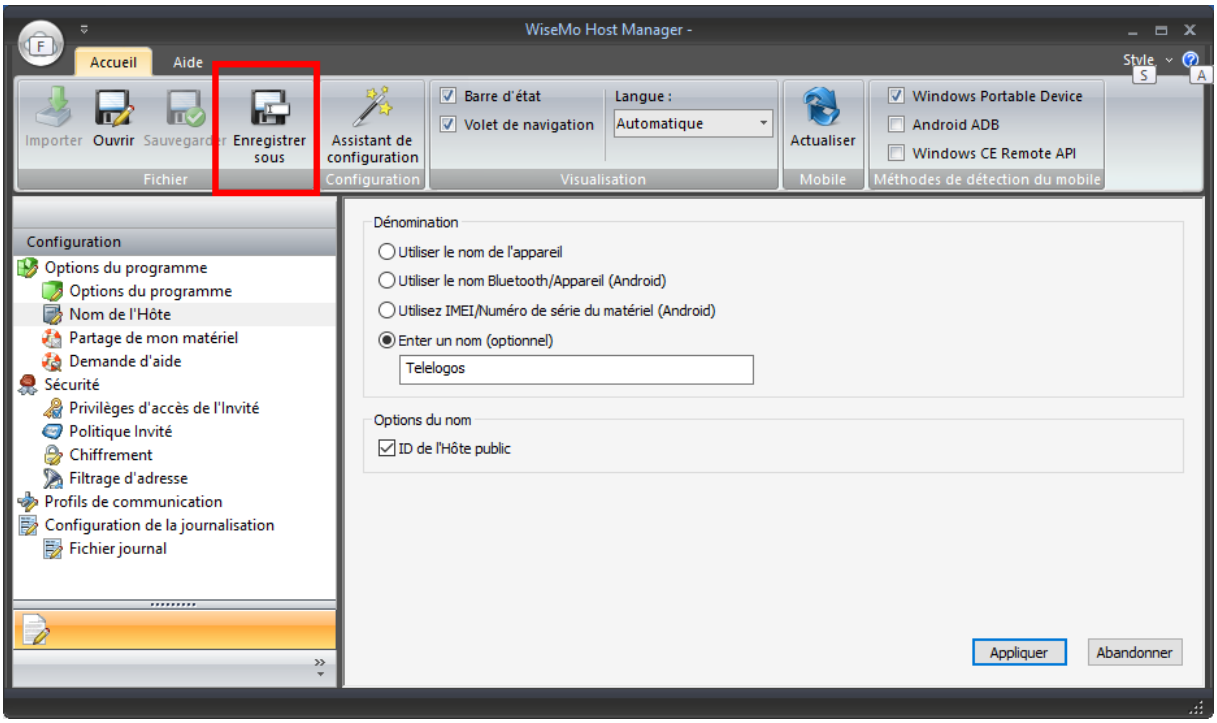

Enregistrer sous
Assistant de configuration
Configuration
Barre
Volet

3 – Modify the configuration to define a specific host name (e.g. Telelogos)



The host name will be automatically customized on each device when the package is installed. It will be replaced by ***<company ID>-<device ID>*** according to the information entered in the CLYD console when creating the package.

4 – Save the file with the name "host.xml".



Creating the CLYD installation package

1 – Add the host.xml file and the WiseMo APK to the CLYD catalog

2 – Create an installation package by integrating:
- The WiseMo APK in the "In-House applications" tab
- The host.xml file in the "Files" tab

Deployment

When deploying the CLYDMediaContact package on the device, the installation program will automatically carry out the actions below:
- Install CLYD
- Copy the host.xml file to the /Wsmhost folder
- Modify the host.xml file (customizing hostname)
- Install and start WiseMo

Chapter 18. I want to perform specific actions on my ZEBRA devices (advanced)

18.1. MX file provisioning

Use the ZEBRA STAGENOW utility to create **Mobility Extension (MX)** profiles in order to customize or add more specific settings to Zebra devices.
You can also use it to prepare an Android version update or the deployment of a security patch.

CLYDMediaContact uses the command "**prov_mx {file_path.xml}**" to apply the .XML files from STAGENOW.

Error code (back from the prov_mx command):

Add-on not present: Error, not Zebra device
MX file missing: Error, no file found
SDK not compatible: Cannot create process prov_mx. (Type: PowerMgr, Error Description: The DSD version is higher than the currently supported DSD 7.2).
Update file missing: Cannot create process prov_mx. (Name: ResetAction, Error Description: Update file not found)

Example of xml zebra file to update an OS (generated with StageNow) :

```
<wap-provisioningdoc>
  <characteristic version="10.1" type="PowerMgr">
  <parm name="ResetAction" value="8" />
  <characteristic type="file-details">
  <parm name="ZipFile" value="/sdcard/HE_FULL_UPDATE_14-23-05.00-UG-U03-STD-HEL-04.zip" />
  </characteristic>
  </characteristic>
</wap-provisioningdoc>
```

18.2. Remote control (WiseMo)

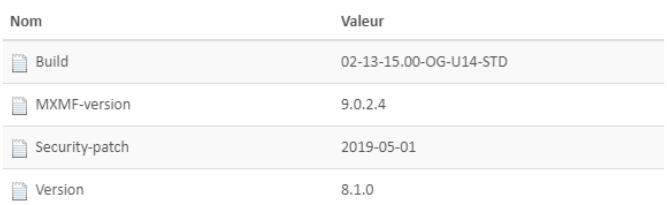
For ZEBRA devices, with Android version 8.1 and MXMF 8.4 minimum.
The WiseMo remote control functions (WiseMo Host v18 minimum) on ZEBRA devices, but requires WiseMo key provisioning on the device.

Provisioning WiseMo keys
Using the WiseMo remote control requires WiseMo keys integration via the provisioning of an MX file. This provisioning is done automatically when installing the CLYDMediaContact client, provided that your package contains the ZEBRA add-on (See: Creating a CLYD package).

18.3. Hardware inventory

Available in the hardware inventory details (operating system):

- The update patch version is part of the Build Number (example: U14 for update 14, see screenshot).
- The security patch version appears as a date with the label "Security-patch".
- The MX version appears with the label "MXMF-Version"

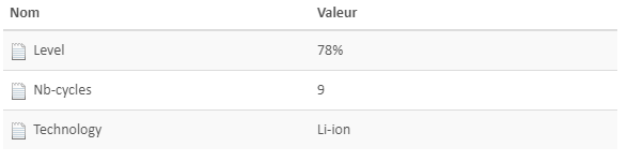


Available in the hardware inventory details (Batteries - main).

Number of battery charging cycles (nb-cycles)

Available on some devices with a "Power Precision" battery: MC40, MC92, TC20, TC25, TC55, TC75, TC70, MC67

ZEBRA URL:



Nom	Valeur
Level	78%
Nb-cycles	9
Technology	Li-ion

Chapter 19. I want to perform specific actions on my Samsung devices (advanced)

Prerequisite : Samsung Android devices.

19.1. I want to manage the update of my Samsung devices (EFOTA)

Samsung's Enterprise Firmware Over the Air (E-FOTA) system lets you control the deployment of updates on your Samsung terminals. For example, you can:

- Wait for updates to be tested and validated before deploying them to your fleet;
- Force updates so that your terminals remain up to date without user action;

To perform these operations, you can export terminals from Clyd and upload them to E-FOTA.

E-FOTA requires devices running Android 7.0 or later and Samsung Knox 2.71 or later. You must also purchase E-FOTA licenses from Samsung or an authorized reseller to use this function.

For more information, visit https://www.samsungknox.com/fr/solutions/it-solutions/samsung_e-fota.

19.2. Specific security parameters

In Clyd, security profiles allow you to apply Samsung specific restrictions like url filtering, buttons blocking… (see 12.2).

19.3. I want to set access point networks on my Samsung devices (APN)

Samsung devices only, requires Telelogos add-on for Samsung).
Use Clyd processes "Command execution" and distribute APN.

prov_apn android.xml

This command allows you to create an APN input. The information is contained in the apn_android.xml file (a template of this file is available in the MediaContact distribution folders). Examples are given below.

The file can contain multiple <entry> items in order to configure multiple APNs in a single command.

prov_apn -d nom_APN

Use this command to delete all APN entries with this name (in Android, multiple APN entries can have the same name).

prov_apn -a nom_APN

Use this command to select the APN entry with this name. If more than one APN entry has this name, the first entry found will be selected.

Examples of APN configuration files:

Base configuration:

```
<?xml version="1.0" encoding="UTF-8"?>
<apns>
    <entry>
        <name>explicit name</name>
        <apn>APN name</apn>
        <mmsc>MMS server URL</mmsc>
        <mcc>Mobile Country Code</mcc>
        <mnc>Mobile Network Code</mnc>
    </entry>
</apns>
```

Advanced configuration:

<authType> can contain the values: -1 (undefined), 0 (none), 1 (PAP), 2 (CHAP), 3 (PAP or CHAP).

<type> can contain the values: default, mms, supl.

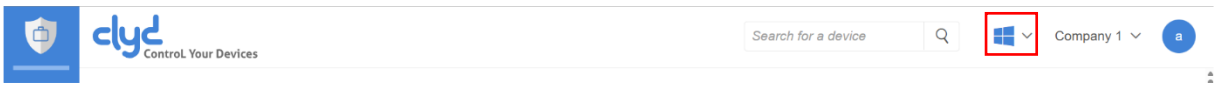
<protocol> can contain the values: IP (default value), IPV6, IPV4V6.

<roamingProtocol> can contain the values: IP (default value), IPV6, IPV4V6.

```
<?xml version="1.0" encoding="UTF-8"?>
<apns>
    <entry>
        <name>explicit name</name>
        <apn>APN name</apn>
        <proxy>proxy address</proxy>
        <port>port number</port>
        <user>user name</user>
        <password>password</password>
        <server>server address</server>
        <mmsc>MMS server URL</mmsc>
        <mmsProxy>MMS proxy address</mmsProxy>
        <mmsPort>MMS port address</mmsPort>
        <mcc>Mobile Country Code</mcc>
        <mnc>Mobile Network Code</mnc>
        <authType>authentication type</authType>
        <type>type of access point</type>
        <protocol>protocol</protocol>
        <mvno_type>MVNO type of the APN</mvno_type>
        <mvno_value>MVNO value</mvno_value>
        <roamingProtocol>roaming protocol</roamingProtocol>
    </entry>
</apns>
```

Chapter 20. I want to manage my Windows Devices (advanced)

CLYD displays your Windows devices. You can access the device information (coordinates, software and hardware inventory, logs). You can define conformity criteria (last connection date, application version) and activate remote control software (WiseMo in LAN or My Cloud mode).



20.1. Installing MediaContact Windows client

First you need to download the Windows client:

- Go to your dedicated portal on the Telelogos website or from the "Clyd Download" page on the Telelogos website.
- If you are using Clyd from your own server, you can download the MediaContact (Clyd) Windows client to install on targeted PCs from C:\MediaContact\MCS\MediaContact Client (launch by running Setup.exe).

You then need to install the client on the targeted Windows devices by running the downloaded "Setup.exe" file.

During installation, you must choose an identifier for the device and enter the identifier of the Clyd company in which you wish to manage this device. You'll find this information in Clyd's Company menu (note that if you've changed your company name, the identifier remains the same as the one you defined when you created your company):

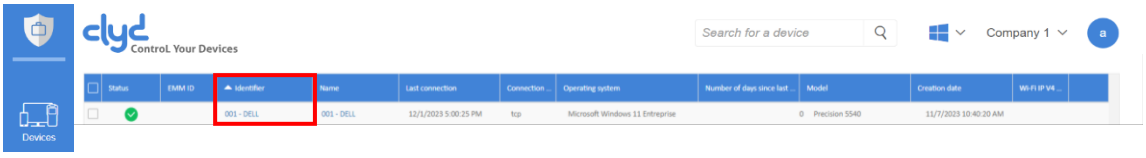
Once the installation is done:

You can check that the MediaContact client is installed on the device by clicking on the MediaContact (Clyd client) icon in the system tray.

To set the connection between client and your server, you need to "Open the MediaContact Client settings window" by right-clicking on the MediaContact client icon, then set the host name and port number and click "OK".

Caution: The host name must correspond to the ip address or url of the server to which the MediaContact client connects. It is essential for establishing the connection between your device and your server. If you have enrolled Android devices in your company, you can find the host name in an installation package in the "Configuration" menu (see 4.2 page 14).

Back on Clyd, you can check in the « Devices » menu that your device appeared (see its identifier in the list):



20.2. Configuration menu

Access rights
Allows account management for access to the CLYD console. (See chapter 15.3 page 73)

Remote control

You can configure the remote-control settings from the CLYD console. (See chapter Chapter 1 page 98)

Device compliance
You can configure the device compliance indicator. The available criteria are the last connection date and inventory application version. The conformity indicator is displayed on the device list and the details of each device. (See chapter 15.6 page 78)

20.3. Device menu

Shows the device list and details of one of the listed devices.

Device details
You can view the device information (coordinates, hardware inventory, software inventory, logs, geolocation coordinates, last connection date)
You can control the device remotely using WiseMo.

Actions on devices.
You can access the actions from the device list and device details. (Unassign a device, delete a device)

20.4. Dashboard menu

You can create a custom dashboard using the plug-in components available for Windows. Three of these components can be displayed on the home page.

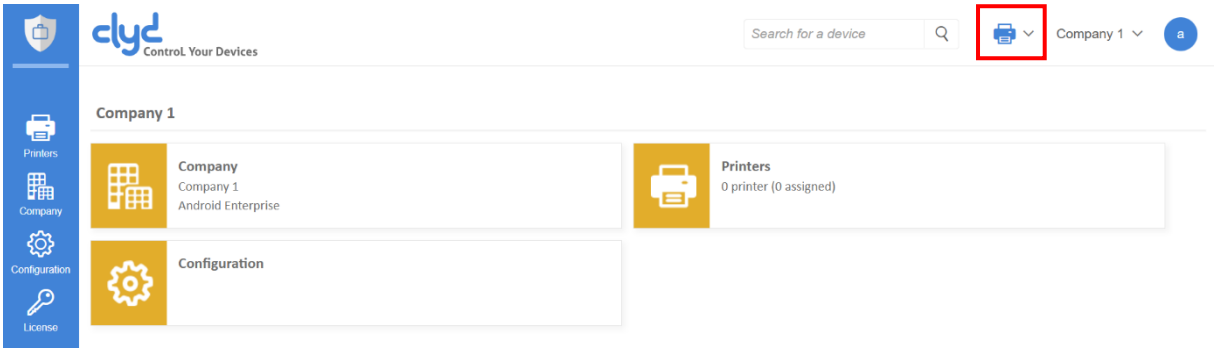
List of components:

- Device compliance
- Device properties (Device coordinates, Brand, Model, Operating system version)
- Application version
- CLYDMediaContact version
- Status of last connections
- Today's connection status
- Status of connections for the last 7 days
- Status of connections for the last 8 hours

Chapter 21. Zebra Printers

Use CLYD to view the configuration of your Zebra printers, provided they support Link OS with the WebLink communication protocol.

Go to the printer home page:

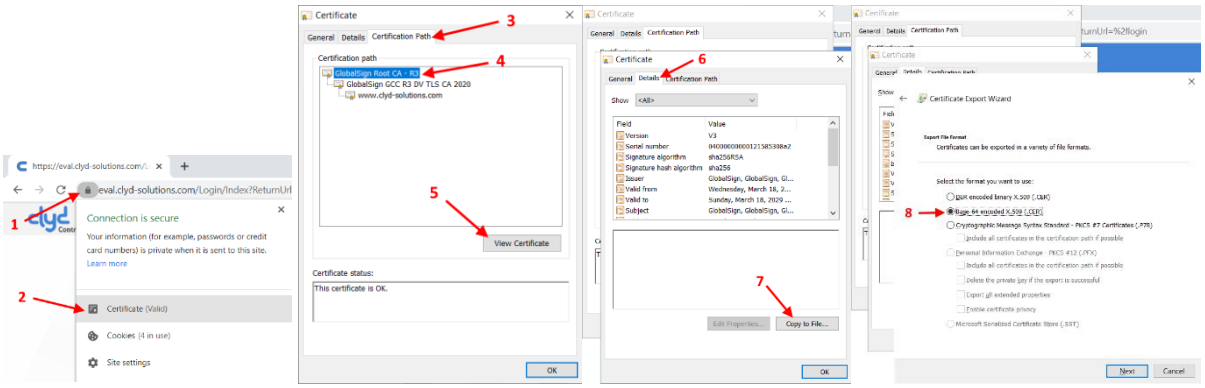


21.1. Registering a printer

To register a printer on the CLYD Server, you must provide the following information:

- The root certificate of the certification authority having delivered the CLYD server's certificate. This certificate will allow the printer to validate the TLS certificate that will be sent back to it each time it connects to the CLYD server.
- The registration URL to the CLYD server.

The root certificate of the certification authority is obtained from the web browser, following the procedure below:



The resulting file contains the root certificate of the certification authority in PEM format. This file must then be added to the printer using the following SGD command, via the Zebra Setup Utilities tool:

{}--ecb22891<CR><LF>
Content-Disposition: filename="E:WEBLINK1_CA.NRD";action="store"<CR><LF>
Content-Type: application/octet-stream<CR><LF>
Content-Transfer-Encoding: binary<CR><LF>
<CR><LF>
<Insert the contents of the downloaded file here, including the tags "BEGIN CERTIFICATE" and "END CERTIFICATE"><CR><LF>
--ecb22891--<CR><LF>

You can obtain the registration URL from the Company menu – Registration tab by clicking on the "Copy" button. It is then copied to the clipboard of your computer. It can be customized, if needed. This URL must be added to the printer using the following SGD command, via the Zebra Setup Utilities tool:

{}{"weblink.ip.conn1.location":"*<Insert the URL copied to the clipboard here>*"}*<CR><LF>*

After performing these two operations, you must restart the printer. This can be done physically on the printer, or by using the following SGD command via the Zebra Setup Utilities tool:

! U1 do "device.reset" ""*<CR><LF>*

After restarting, the printer will automatically connect to the CLYD Server and will appear in the company's printer list.

21.2. Printer menu

Shows the printer list and details of one of the listed printers.

List of printers
The company's printers appear in the list of printers. In addition to the general columns, you can add any inventory data as a column in the list.

Printer details
You can view the printer information (coordinates, hardware inventory).

Actions on printers.
You can access the actions from the printer list and printer details:

- Run a command
- Request a hardware inventory
- Delete a printer

Run a command
Selecting this menu displays a window in which you have to type or paste the commands to be executed. Clyd supports the different languages offered by Zebra:

- ZPL commands (Zebra Programming Language) ;
- SGD commands (Set Get Do) :
    - In plain text format ;
    - In JSON format.

Please refer to Zebra documentation for more information about these languages.

21.3. Configuration – Scheduling hardware inventories

A first hardware inventory is systematically uploaded when the printer is registered.

It is also possible to schedule a regular inventory report for all the printers of a company by using the inventory schedule in the "Configuration" menu. It is possible to configure several settings:

Call frequency
The frequency can be expressed in Minutes, Hours or Days.

Validity periods
Allows you to reduce connections for specific time periods or days of the week.

By default, no inventory schedule is defined.

Warning:

The following API has been deprecated. It no longer allows you to request an inventory for all printers on the server and now returns the status HTTP 403:

Method	**URL**
POST	**https://{server_address}/clyd-iot/api/v1/printers/zebra/inventory**

Response status:

HTTP status	**Description**
403 Forbidden	This API is no longer available.

Chapter 22. I want to view or set historicization parameters (advanced)

22.1. I want to view Clyd logs for my devices

The "Logs" menu in the Clyd console lets you find Clyd events applied to your devices.
A menu at the top right allows you to export the list displayed, update the list and apply specific filters.
Events relating to companies or devices are referenced in this view. The history of a specific device is available in its device file.

The following types of events are logged and stored for 15 days:

- Application events
- Lifecycle changes
- Provisioning/Monitoring
- Kiosk/ProSpace
- oEMM profiles
- Remote control
- Processes
- System

22.2. I want to set the CLYD Server historicization

The Nlog severity level is defined in the WEB.CONFIG file (Nlog is the library used in CLYD). By default the Nlog severity level is set to WARN. Support can modify the severity level. The W3SVC service must be restarted for the changes to be applied.

WEB.CONFIG file tag:

```
<rules>
  <logger name="*" minlevel="Warn" writeTo="logfile" />
</rules>
```

Severity levels table:

Level	Example
Fatal	Highest level: important stuff down
Error	For example, application crashes / exceptions.
Warn	Incorrect behavior but the application can continue
Info	Normal behavior like mail sent, user updated profile, etc.
Debug	Executed queries, user authenticated, session expired
Trace	Begin method X, end method X, etc.

CLYD web console log:
This is the virtual folder /Default Web Site/CLYD that logs the usage of graphical interface
Location of the WEB.CONFIG file: C:\Program Files (x86)\Telelogos\CLYD**web.config**
Location of the log file: C:\MediaContact\MCS\MediaContact\Tmp\Logs\Clyd**console.log**

Web services log:
This is the virtual folder /Default Web Site/MediaContactWebServices that logs the usage of Web services (kiosk status, software inventory, etc.)
Location of the WEB.CONFIG file: C:\Program Files (x86)\Telelogos\MediaContact Web Services **web.config**
Location of the log file: C:\MediaContact\MCS\MediaContact\Tmp\Logs\mcws**mcws.log**

Android Enterprise log
Contains information about requests sent to the Google servers.
Location of the NLOG.CONFIG file: C:\Program Files (x86)\Telelogos\CLYD\bin**ManageAppEMM.exe.config**
Location of the log file: C:\MediaContact\MCS\MediaContact\Tmp\Logs\ManageAppEMM**ManageAppEMM.log**

Chapter 23. Managing CLYD using the web API

Some CLYD features can be managed via web APIs. This allows a desktop application to access certain CLYD features. Calling these APIs via a webpage script is not supported.

These web APIs, in REST format, require authentication. Only a console administrator attached to a company can authenticate themselves to use these APIs

Caution: An account used to connect to Clyd via API can no longer connect to the interface (for security reasons).

Your API calls must always target a url starting with the url of your server. For example, if you connect to Clyd at https://eval.clyd-solutions.com/, all your API requests must start with https://eval.clyd-solutions.com/ and end with the specific url described after for each method.

23.1. Authentication API

This API allows the calling application to authenticate itself to the CLYD server according to the following rules:

- Authentication is done using the same accounts as those used to log in to the CLYD console.
- Only administrator accounts are authorized to authenticate themselves via API.
- Global administrator accounts are not authorized to authenticate themselves via API.

Method	URL
POST	**/public/api/v{version}/Authentication/Authenticate**

Command line settings:

Settings	Description	Value
{version}	API version to use	1

Request headers:

Header	Content
Content-Type	application/json

Request body:
Character string in JSON format containing the authentication information:

```
{
  "companyName": "string",
  "userName": "string",
  "password": "string"
}
```

Company name (required)
Administrator user name (required)
User password (required)

Response status:

HTTP status	Description
200 Success	The user is correctly authenticated with the CLYD server. Content is available in the body of the response.
400 Bad request	At least one of the mandatory attributes of the request body is empty or missing.
	The JSON of the request body is syntactically incorrect.
	The API version provided as a parameter is incorrect.
401 Unauthorized	The information provided in the request body does not allow the user to be authenticated with the CLYD server.

	The "Authorization" request header contains an incorrect or expired authentication token. **The "Authorization" request header should not be filled in for this API.**
404 Not Found	The URL is incorrectly formatted.
405 Method Not Allowed	The request was sent using an incorrect method.
	The URL is incorrectly formatted.
415 Unsupported Media Type	The "Content-Type" request header is missing or does not indicate "application/json".

Response body (if HTTP 200 status):
Character string containing the authentication token between quotes.

23.2. API to retrieve the list of devices

This API allows the calling application to retrieve the list of company devices and to access the main properties of each device: contact details, status, brand, model, IMEI number, serial number, life cycle, IP parameters.

Method	URL
GET	**/public/api/v{version}/Devices/{typeOS}**

Supported device types:
All

Command line settings:

Settings	Description	Values
{version}	API version to use	1
{typeOS}	Type of devices that the API must return	android windows printer all

Request headers:

Header	Content
Authorization	"Bearer" keyword followed by a space, and then the authentication token returned by the authentication API (without quotes)
Accept	application/json

Request body:
None

Response status:

HTTP status	Description
200 Success	The request executed successfully. Content is available in the body of the response.
400 Bad Request	The API version provided as a parameter is incorrect.
401 Unauthorized	"Authorization" request header contains an incorrect or expired authentication token, or has no token.

404 Not Found	The URL is incorrectly formatted.
	The type of device provided as a parameter is incorrect.
405 Method Not Allowed	The request was sent using an incorrect method.
	The URL is incorrectly formatted.

Response body (if HTTP 200 status):
Character string in JSON format containing a device table:

```
[
  {
    "deviceType": "string",
    "identifier": "string",
    "name": "string",
    "firstName": "string",
    "street": "string",
    "city": "string",
    "postCode": "string",
    "country": "string",
    "email": "string",
    "fixedPhone": "string",
    "userFunction": "string",
    "service": "string",
    "location": "string",
    "state": "string",
    "mobilePhone": "string",
    "brand": "string",
    "model": "string",
    "imei": "string",
    "serialNumber": "string",
    "ipAddress": "string",
    "wifiIpv4Address": "string",
    "wifiMacAddress": "string",
    "status": integer,
    "lifeCycle": {
      "status": "string",

      "comment": "string"
    }
  },
  ...
]
```

Type ("android", "windows"," printer")
Identifier
Name
First name
Address
City
Zip code
Country
Email
Landline telephone
Function
Service
Location
State/Province
Mobile phone
Brand
Model
IMEI (null for printer type)
Serial number
IP address (null for printer type)
Wi-Fi IP V4 address (null for printer type)
Wi-Fi MAC address (null for printer type)
Status *
Life cycle (null for printer type)
Device status ("none", "production"," stock", "maintenance", "lost", "non-repairable")
Comment

* Status:

For Android, Windows, Windows CE/Mobile: 0 = unassigned, 1 = suspended, 2 = not compliant, 3 = compliant

For Zebra printers: 0 = unassigned, 1 = suspended, 3 = assigned

23.3. API to import devices (creation and/or modification)

This API allows the calling application to import devices into the Clyd directory using a CSV file.

Method	URL
POST	**/public/api/v{version}/Devices/{typeOS}/Import**

Supported device types:
All

Command line settings:

Parameter	Description	Values
{version}	API version to use	1
{typeOS}	Type of devices that the API must import	android windows printer

Request headers:

Header	Content
Authorization	"Bearer" keyword followed by a space, and then the authentication token returned by the authentication API (without quotes)
Content-Type	multipart/form-data; boundary=xxxxxxxxxxx where xxx represents a string containing up to 70 printable characters with an ASCII code less than 128.
Accept	application/json

Request body:

xxxxxxxxxxx *(string corresponding to the value of "boundary" in the Content-Type)*
Content-Disposition: form-data; name=""; filename=" *file_name*.csv"
Content-Type: text/csv
File content *
xxxxxxxxxxx *(string corresponding to the value of "boundary" in the Content-Type)*
Content-Disposition: form-data; name="importType"
Import type **
xxxxxxxxxxx-- *(string corresponding to the value of "boundary" in the Content-Type, followed by two hyphens)*

* CSV file content: please refer to chapter 8.3 for a description of the expected format

** Import type:

- "create" = creation of devices;
- "update" = modification of devices;
- "create_and_update" : creation and modification of devices.

Response status:

HTTP status	Description
200 Success	The request executed successfully. Content is available in the body of the response.
400 Bad request	At least one of the mandatory attributes of the request body is empty or missing.
	The JSON of the request body is syntactically incorrect.
	The API version provided as a parameter is incorrect.
401 Unauthorized	"Authorization" request header contains an incorrect or expired authentication token, or has no token.
404 Not Found	The URL is incorrectly formatted.
	The import type provided as a parameter is incorrect.
405 Method Not Allowed	The request was sent using an incorrect method.
	The URL is incorrectly formatted.
406 Not Acceptable	A column is duplicated or the column 'identifier' is missing in the CSV file.
415	The "Content-Type" request header is missing or does not indicate "multipart/form-data".

Unsupported Media Type	The file sent is not a CSV file.

Response body (if HTTP 200 status):
Character string in JSON format containing import execution report:

```
{
  "report": {
    "addedDevices": { "count": entier },
    "updatedDevices": { "count": entier },
    "ignoredDevices": { "count": entier },
    "linesInError": {
      "count": entier,
      "details": [
        {
          "lineNumber": entier,
          "identifier": "string",
          "errors": [
              "string",
              ...
          ]
        },
        ...
      ]
  }
}
```

Import exécution report
Number of created devices
Number of modified devices
Number of ignored devices
Failed lines
Number of failed lines
Failed lines details

Number of the failed line in the CSV file
Identifier of failed device
Error list
Error description

23.4. API to customize device properties

This API allows the calling application to customize certain properties of the company's devices: contact details, life cycle.

Method	URL
PATCH	**/public/api/v{version}/Devices/{identifier}**

Supported device types:
All

Command line settings:

Settings	Description	Value
{version}	API version to use	1
{identifier}	Device identifier	

Request headers:

Header	Content
Authorization	"Bearer" keyword followed by a space, and then the authentication token returned by the authentication API (without quotes)
Content-Type	application/json

Request body:
String in JSON format containing the properties to be modified. Properties not included in the request will not be modified on the CLYD server. The name cannot be empty. The syntax of the email is checked.

```
{
  "name": "string",
  "firstName": "string",
  "street": "string",
```

Name (required)
First name
Address

```
  "city": "string",
  "postCode": "string",
  "country": "string",
  "email": "string",
  "fixedPhone": "string",
  "userFunction": "string",
  "service": "string",
  "location": "string",
  "state": "string",
  "mobilePhone": "string",
  "lifeCycle": {
   "status": "string",

   "comment": "string"
  }
}
```

City
Zip code
Country
Email
Landline telephone
Function
Service
Location
State/Province
Mobile phone
Life cycle (ignored for printer type)
Device status ("none", "production", "stock", "maintenance"," lost"," non-repairable")
Comment

Response status:

HTTP status	Description
204 No Content	The request executed successfully. No content is available in the body of the response.
400 Bad request	At least one of the mandatory attributes of the request body is empty or missing.
	The JSON of the request body is syntactically incorrect.
	The API version provided as a parameter is incorrect.
401 Unauthorized	"Authorization" request header contains an incorrect or expired authentication token, or has no token.
404 Not Found	The URL is incorrectly formatted.
	The device identifier provided as a parameter is incorrect.
405 Method Not Allowed	The request was sent using an incorrect method.
	The URL is incorrectly formatted.
415 Unsupported Media Type	The "Content-Type" request header is missing or does not indicate "application/json".

Response body:
None

23.5. API to retrieve conformity information from a device

This API allows the calling application to retrieve detailed information about the conformity of the device with the conformity rules specified in the CLYD server configuration.

Method	URL
GET	**/public/api/v{version}/Devices/{identifier}/Compliance**

Supported device types:
Android, Windows, Windows CE/Mobile

Command line settings:

Settings	Description	Value
{version}	API version to use	1
{identifier}	Device identifier	

Request headers:

Header	Content
Authorization	"Bearer" keyword followed by a space, and then the authentication token returned by the authentication API (without quotes)
Accept	application/json

Request body:
None

Response status:

HTTP status	Description
200 Success	The request executed successfully. Content is available in the body of the response.
400 Bad request	The API version provided as a parameter is incorrect.
401 Unauthorized	"Authorization" request header contains an incorrect or expired authentication token, or has no token.
403 Forbidden	The API does not support the type of the device whose identifier is provided in the request.
404 Not Found	The URL is incorrectly formatted.
	The device identifier provided as a parameter is incorrect.
405 Method Not Allowed	The request was sent using an incorrect method.
	The URL is incorrectly formatted.

Response body (if HTTP 200 status):
Character string in JSON format containing the compliance status and the table of non-compliant criteria:

```
{
  "status": entier,
  "nonCompliantCriteria": [
   {
    "type": "string",
    "rule": "string",
    "evaluation": "string"
   },
   ...
  ]
}
```

Compliance status *
Non-compliant criteria (if status = 2):

Type of criteria **
Compliance rule
Evaluation of device non-compliance

* Compliance status: 0 = unassigned, 1 = suspended, 2 = not compliant, 3 = compliant

** Type of criteria:

- ClydMediaContactVersion: CLYDMediaContact version installed on the device (Android, Windows)
- LastConnectionDate: last connection of the device to the server (Android, Windows, Windows CE/Mobile)
- KioskVersion: kiosk version activated on the device (Android)
- KioskActivation: kiosk activated or not on the device (Android)
- ApplicationVersion: application version installed on the device (Android, Windows)
- ApplicationPresence: application installed or not on the device (Windows CE/Mobile)
-

23.6. API to ask an Android device to connect

This API allows the calling application to ask an Android device to connect to the server. All modified data associated with the device (monitoring profiles, provisioning profiles, kiosks, WorkSpaces, EMM profiles, etc.) is then prepared by the server, which subsequently sends the device a connection request. This connection request is made by sending an FCM message (requires a Google ID to be associated with the device). The next time the device connects, it will retrieve its entire configuration.

This API has the same behavior as the "Connect" menu available in the list of devices and in a device's information sheet.

Method	URL
POST	**/public/api/v{version}/Devices/{identifier}/Connect**

Supported device types:
Android

Command line settings:

Settings	Description	Value
{version}	API version to use	1
{identifier}	Device identifier	

Request headers:

Header	Content
Authorization	"Bearer" keyword followed by a space, and then the authentication token returned by the authentication API (without quotes)

Request body:
None

Response status:

HTTP status	Description
204 No Content	The request executed successfully. No content is available in the body of the response.
400 Bad request	The API version provided as a parameter is incorrect.
401 Unauthorized	"Authorization" request header contains an incorrect or expired authentication token, or has no token.
403 Forbidden	The API does not support the type of the device whose identifier is provided in the request.
404 Not Found	The URL is incorrectly formatted.
	The device identifier provided as a parameter is incorrect.
405 Method Not Allowed	The request was sent using an incorrect method.
	The URL is incorrectly formatted.

Response body:
None

23.7. API to fully reload an Android device with a device call

This API allows the calling application to fully reload an Android device. All data associated with the device (monitoring profiles, provisioning profiles, kiosks, WorkSpaces, EMM profiles, etc.) is then prepared by the

server, which subsequently sends the device a connection request. This connection request is made by sending an FCM message (requires a Google ID to be associated with the device). The next time the device connects, it will retrieve its entire configuration.

This API has the same behavior as the "Reload" menu available in the list of devices and in a device's information sheet.

Method	URL
POST	**/public/api/v{version}/Devices/{identifier}/Synchronize**

Supported device types:
Android

Command line settings:

Settings	Description	Value
{version}	API version to use	1
{identifier}	Device identifier	

Request headers:

Header	Content
Authorization	"Bearer" keyword followed by a space, and then the authentication token returned by the authentication API (without quotes)

Request body:
None

Response status:

HTTP status	Description
204 No Content	The request executed successfully. No content is available in the body of the response.
400 Bad request	The API version provided as a parameter is incorrect.
401 Unauthorized	"Authorization" request header contains an incorrect or expired authentication token, or has no token.
403 Forbidden	The API does not support the type of the device whose identifier is provided in the request.
404 Not Found	The URL is incorrectly formatted.
	The device identifier provided as a parameter is incorrect.
405 Method Not Allowed	The request was sent using an incorrect method.
	The URL is incorrectly formatted.

Response body:
None

23.8. API to delete user data from applications on an Android device

This API allows the calling application to send a request to the device to delete user data from certain applications. This request is made by sending an FCM message (requires a Google ID to be associated with the device).

Method	URL

POST	/public/api/v{version}/Devices/{identifier}/ClearApplicationUserData

Supported device types:
Android

Command line settings:

Settings	Description	Value
{version}	API version to use	1
{identifier}	Device identifier	

Request headers:

Header	Content
Authorization	"Bearer" keyword followed by a space, and then the authentication token returned by the authentication API (without quotes)
Content-Type	application/json

Request body:
Character string in JSON format containing the relevant applications:

```
{
  "packages": [
   "string",
   ...
  ]
}
```

Table containing the package names of the applications for which the user data must be deleted (required). The following package names are forbidden: com.telelogos.mediacontact and com.telelogos.clyddpc

Response status:

HTTP status	Description
204 No Content	The request executed successfully. No content is available in the body of the response. This status ensures that data deletion requests have been sent to the device, but does not ensure that the device has successfully performed these operations.
400 Bad request	The API version provided as a parameter is incorrect.
	At least one of the mandatory attributes of the request body is empty or missing.
	The JSON of the request body is syntactically incorrect.
401 Unauthorized	"Authorization" request header contains an incorrect or expired authentication token, or has no token.
403 Forbidden	The API does not support the type of the device whose identifier is provided in the request.
	The device concerned by the request cannot be reached because it does not have a Google ID.
	The package attribute contains one of the forbidden package names: com.telelogos.mediacontact or com.telelogos.clyddpc
404 Not Found	The URL is incorrectly formatted.
	The device identifier provided as a parameter is incorrect.
405 Method Not Allowed	The request was sent using an incorrect method.
	The URL is incorrectly formatted.
415	The "Content-Type" request header is missing or does not indicate "application/json".

Unsupported Media Type	

Response body:
None

23.9. API to broadcast a message to an Android device

This API allows the calling application to send the device a message to be displayed. This request is made by sending an FCM message (requires a Google ID to be associated with the device).

This API has the same behavior as the "Broadcast a message" menu available in the list of devices and in a device's information sheet.

Method	URL
POST	**/public/api/v{version}/Devices/{identifier}/BroadcastMessage**

Supported device types:
Android

Command line settings:

Settings	Description	Value
{version}	API version to use	1
{identifier}	Device identifier	

Request headers:

Header	Content
Authorization	"Bearer" keyword followed by a space, and then the authentication token returned by the authentication API (without quotes)
Content-Type	application/json

Request body:
Character string in JSON format containing the relevant applications:

```
{
  "title": "string",
  "message": "string"
}
```

Message title. By default: "Message Clyd" (FR) or "Clyd Message" (EN)
Message (required)

Response status:

HTTP status	Description
204 No Content	The request executed successfully. No content is available in the body of the response. This status ensures that the message broadcast request has been sent to the device, but does not ensure that the device has displayed the message.
400 Bad request	The API version provided as a parameter is incorrect.
	At least one of the mandatory attributes of the request body is empty or missing.
	The JSON of the request body is syntactically incorrect.
401 Unauthorized	"Authorization" request header contains an incorrect or expired authentication token, or has no token.

403 Forbidden	The API does not support the type of the device whose identifier is provided in the request.
	The device concerned by the request cannot be reached because it does not have a Google ID.
404 Not Found	The URL is incorrectly formatted.
	The device identifier provided as a parameter is incorrect.
405 Method Not Allowed	The request was sent using an incorrect method.
	The URL is incorrectly formatted.
415 Unsupported Media Type	The "Content-Type" request header is missing or does not indicate "application/json".

Response body:
None

23.10. API to reboot an Android device

This API allows the calling application to send a reboot request to the device. This request is made by sending an FCM message (requires a Google ID to be associated with the device).

This API has the same behavior as the "Reboot" menu available in the list of devices and in a device's information sheet.

Method	URL
POST	**/public/api/v{version}/Devices/{identifier}/Reboot**

Supported device types:
Android

Command line settings:

Settings	Description	Value
{version}	API version to use	1
{identifier}	Device identifier	

Request headers:

Header	Content
Authorization	"Bearer" keyword followed by a space, and then the authentication token returned by the authentication API (without quotes)

Request body:
None

Response status:

HTTP status	Description
204 No Content	The request executed successfully. No content is available in the body of the response. This status ensures that the reboot request has been sent to the device, but does not ensure that the device has rebooted.
400 Bad request	The API version provided as a parameter is incorrect.
401 Unauthorized	"Authorization" request header contains an incorrect or expired authentication token, or has no token.

403 Forbidden	The API does not support the type of the device whose identifier is provided in the request.
	The device concerned by the request cannot be reached because it does not have a Google ID.
404 Not Found	The URL is incorrectly formatted.
	The device identifier provided as a parameter is incorrect.
405 Method Not Allowed	The request was sent using an incorrect method.
	The URL is incorrectly formatted.

Response body:
None

23.11. API to make an Android device ring

This API allows the calling application to send a ring request to the device. This request is made by sending an FCM message (requires a Google ID to be associated with the device).

This API has the same behavior as the "Ring" menu available in the list of devices and in a device's information sheet.

Method	URL
POST	**/public/api/v{version}/Devices/{identifier}/Ring**

Supported device types:
Android

Command line settings:

Settings	Description	Value
{version}	API version to use	1
{identifier}	Device identifier	

Request headers:

Header	Content
Authorization	"Bearer" keyword followed by a space, and then the authentication token returned by the authentication API (without quotes)

Request body:
None

Response status:

HTTP status	Description
204 No Content	The request executed successfully. No content is available in the body of the response. This status ensures that the ring request has been sent to the device, but does not ensure that the device has rung.
400 Bad request	The API version provided as a parameter is incorrect.
401 Unauthorized	"Authorization" request header contains an incorrect or expired authentication token, or has no token.
403 Forbidden	The API does not support the type of the device whose identifier is provided in the request.

	The device concerned by the request cannot be reached because it does not have a Google ID.
404 Not Found	The URL is incorrectly formatted.
	The device identifier provided as a parameter is incorrect.
405 Method Not Allowed	The request was sent using an incorrect method.
	The URL is incorrectly formatted.

Response body:
None

23.12. API to lock an Android device

This API allows the calling application to send a lock request to the device. This request is made by sending an FCM message (requires a Google ID to be associated with the device).

This API has the same behavior as the "Lock" menu available in the list of devices and in a device's information sheet.

Method	URL
POST	**/public/api/v{version}/Devices/{identifier}/Lock**

Supported device types:
Android

Command line settings:

Settings	Description	Value
{version}	API version to use	1
{identifier}	Device identifier	

Request headers:

Header	Content
Authorization	"Bearer" keyword followed by a space, and then the authentication token returned by the authentication API (without quotes)

Request body:
None

Response status:

HTTP status	Description
204 No Content	The request executed successfully. No content is available in the body of the response. This status ensures that the lock request has been sent to the device, but does not ensure that the device has been locked.
400 Bad request	The API version provided as a parameter is incorrect.
401 Unauthorized	"Authorization" request header contains an incorrect or expired authentication token, or has no token.
403 Forbidden	The API does not support the type of the device whose identifier is provided in the request.
	The device concerned by the request cannot be reached because it does not have a Google ID.
404 Not Found	The URL is incorrectly formatted.
	The device identifier provided as a parameter is incorrect.
405 Method Not Allowed	The request was sent using an incorrect method.
	The URL is incorrectly formatted.

Response body:
None

23.13. API to unlock an Android device

This API allows the calling application to send an unlock request to the device. This request is made by sending an FCM message (requires a Google ID to be associated with the device).

This API has the same behavior as the "Unlock" menu available in the list of devices and in a device's information sheet.

Method	URL
POST	**/public/api/v{version}/Devices/{identifier}/Unlock**

Supported device types:
Android

Command line settings:

Settings	Description	Value
{version}	API version to use	1
{identifier}	Device identifier	

Request headers:

Header	Content
Authorization	"Bearer" keyword followed by a space, and then the authentication token returned by the authentication API (without quotes)

Request body:
None

Response status:

HTTP status	Description
204 No Content	The request executed successfully. No content is available in the body of the response. This status ensures that the unlock request has been sent to the device, but does not ensure that the device has been unlocked.
400 Bad request	The API version provided as a parameter is incorrect.
401 Unauthorized	"Authorization" request header contains an incorrect or expired authentication token, or has no token.
403 Forbidden	The API does not support the type of the device whose identifier is provided in the request.
	The device concerned by the request cannot be reached because it does not have a Google ID.
404 Not Found	The URL is incorrectly formatted.
	The device identifier provided as a parameter is incorrect.
405 Method Not Allowed	The request was sent using an incorrect method.
	The URL is incorrectly formatted.

Response body:
None

23.14. API to stop the kiosk on an Android device

This API allows the calling application to send the device a request to stop the kiosk. This request is made by sending an FCM message (requires a Google ID to be associated with the device).

This API has the same behavior as the "Stop kiosk" menu available in the list of devices in a device's information sheet.

Method	URL
POST	**/public/api/v{version}/Devices/{identifier}/StopKiosk**

Supported device types:
Android in Dedicated Device or Device Admin company

Command line settings:

Settings	Description	Value
{version}	API version to use	1
{identifier}	Device identifier	

Request headers:

Header	Content
Authorization	"Bearer" keyword followed by a space, and then the authentication token returned by the authentication API (without quotes)

Request body:
None

Response status:

HTTP status	Description
204 No Content	The request executed successfully. No content is available in the body of the response. This status ensures that the request to stop the kiosk has been sent to the device but does not ensure that the kiosk has been stopped.
400 Bad request	The API version provided as a parameter is incorrect.
401 Unauthorized	"Authorization" request header contains an incorrect or expired authentication token, or has no token.
403 Forbidden	The API does not support the type of the device whose identifier is provided in the request.
	The device concerned by the request cannot be reached because it does not have a Google ID.
	The API does not support the type of company in which it is authenticated.
404 Not Found	The URL is incorrectly formatted.
	The device identifier provided as a parameter is incorrect.
405 Method Not Allowed	The request was sent using an incorrect method.
	The URL is incorrectly formatted.

Response body:
None

23.15. API to start the kiosk on an Android device

This API allows the calling application to send the device a request to start the kiosk. This request is made by sending an FCM message (requires a Google ID to be associated with the device).

This API has the same behavior as the "Start kiosk" menu available in the list of devices and in a device's information sheet.

Method	URL
POST	**/public/api/v{version}/Devices/{identifier}/StartKiosk**

Supported device types:
Android in Dedicated Device or Device Admin company

Command line settings:

Settings	Description	Value
{version}	API version to use	1
{identifier}	Device identifier	

Request headers:

Header	Content
Authorization	"Bearer" keyword followed by a space, and then the authentication token returned by the authentication API (without quotes)

Request body:
None

Response status:

HTTP status	Description
204 No Content	The request executed successfully. No content is available in the body of the response. This status ensures that the request to start the kiosk has been sent to the device but does not ensure that the kiosk has been started.
400 Bad request	The API version provided as a parameter is incorrect.
401 Unauthorized	"Authorization" request header contains an incorrect or expired authentication token, or has no token.
403 Forbidden	The API does not support the type of the device whose identifier is provided in the request.
	The device concerned by the request cannot be reached because it does not have a Google ID.
	The API does not support the type of company in which it is authenticated.
404 Not Found	The URL is incorrectly formatted.
	The device identifier provided as a parameter is incorrect.
405 Method Not Allowed	The request was sent using an incorrect method.
	The URL is incorrectly formatted.

Response body:
None

23.16. API to run a server process

This API allows the calling application to run a server process, in normal mode or recovery mode. The recovry mode allows the process to be replayed for the devices of the target for which the previous execution failed.

For a CLYD process, this API has the same behavior as the manual running of processes via the CLYD console.

Method	URL
POST	**/public/api/v{version}/Processes/{processName}/Start**[?mediaContactProcess={mcProcess}] [&recoveryMode={recoveryMode}]

Command line settings:

Settings	Description	Value
{version}	API version to use	1
{processName}	CLYD or MediaContact process name	
{mcProcess}	Indicates whether the process indicated in the {processName} parameter is a process defined in the CLYD console (false) or in the MediaContact console (true)	false *(default)* true
{recoveryMode}	Indicates whether the process indicated in the {processName} parameter must be executed in normal mode (false) or in recovery mode (true)	false *(default)* true

Request headers:

Header	Content
Authorization	"Bearer" keyword followed by a space, and then the authentication token returned by the authentication API (without quotes)

Request body:
None

Response status:

HTTP status	Description
204 No Content	The request executed successfully. No content is available in the body of the response. This status ensures that the process has started, but does not ensure that it has run successfully.
400 Bad request	The API version provided as a parameter is incorrect.
401 Unauthorized	"Authorization" request header contains an incorrect or expired authentication token, or has no token.
403 Forbidden	The API failed to start the process.
	The precedent reques executed successfuly, recover mode not available..
404 Not Found	The URL is incorrectly formatted.
	The name of the process provided as a parameter is incorrect.
405 Method Not Allowed	The request was sent using an incorrect method.
	The URL is incorrectly formatted.

Response body:
None

23.17. API to stop the execution of a server process

This API allows the calling application to stop the execution of a server process.

For a CLYD process, this API has the same behavior as the manual stopping of processes via the CLYD console.

Method	URL
POST	**/public/api/v{version}/Processes/{processName}/Stop**[?mediaContactProcess={mcProcess}]

Command line settings:

Settings	Description	Value
{version}	API version to use	1
{processName}	CLYD or MediaContact process name	
{mcProcess}	Indicates whether the process indicated in the {processName} parameter is a process defined in the CLYD console (false) or in the MediaContact console (true)	false *(default)* true
{recoveryMode}	Indicates whether the process indicated in the {processName} parameter must be executed in normal mode (false) or in recovery mode (true)	false *(default)* true

Request headers:

Header	Content
Authorization	"Bearer" keyword followed by a space, and then the authentication token returned by the authentication API (without quotes)

Request body:
None

Response status:

HTTP status	Description
204 No Content	The request executed successfully. No content is available in the body of the response.
400 Bad request	The API version provided as a parameter is incorrect.
401 Unauthorized	"Authorization" request header contains an incorrect or expired authentication token, or has no token.
403 Forbidden	The API failed to start the process.
404 Not Found	he URL is incorrectly formatted.
	The name of the process provided as a parameter is incorrect.
405 Method Not Allowed	he request was sent using an incorrect method.
	he request was sent using an incorrect method.
	The URL is incorrectly formatted.

Response body:
None

23.18. API to add a Play Store application to the Android catalog

This API allows the calling application to add an application from the Google Play Store to the company's enterprise store.

Method	URL
POST	**/public/api/v{version}/Catalogs/{typeOS}/PlayStoreApplications**

Supported contexts:
Android Enterprise company

Command line settings:

Settings	Description	Value
{version}	API version to use	1
{typeOS}	Context	android

Request headers:

Header	Content
Authorization	"Bearer" keyword followed by a space, and then the authentication token returned by the authentication API (without quotes)
Content-Type	application/json

Request body:
String in JSON format containing the information of the application to be added.

```
{
  "packageName": "string"                    Package name (required)
}
```

Response status:

HTTP status	Description
201 Created	The request executed successfully. No content is available in the body of the response.
400 Bad request	At least one of the mandatory attributes of the request body is empty or missing.
	The JSON of the request body is syntactically incorrect.
	The API version provided as a parameter is incorrect.
401 Unauthorized	"Authorization" request header contains an incorrect or expired authentication token, or has no token.
403 Forbidden	The API does not support the context provided in the request.
	The API does not support the type of company in which it is authenticated.
404 Not Found	The URL is incorrectly formatted.
	The context provided as a parameter is incorrect.
	The package name provided does not exist in the Google Play Store.
405 Method Not Allowed	The request was sent using an incorrect method.
	The URL is incorrectly formatted.
409 Conflict	The application already exists in the catalog.
415 Unsupported Media Type	The "Content-Type" request header is missing or does not indicate "application/json".

Response body:
None

23.19. API to add an in-house application to the Android catalog

This API allows the calling application to add an in-house application (APK) to the company's in-house catalog.

Method	URL
POST	**/public/api/v{version}/Catalogs/{typeOS}/InHouseApplications**

Supported contexts:
Android

Command line settings:

Settings	Description	Value
{version}	API version to use	1
{typeOS}	Context	android

Request headers:

Header	Content
Authorization	"Bearer" keyword followed by a space, and then the authentication token returned by the authentication API (without quotes)
Content-Type	multipart/form-data; boundary=xxxxxxxxxxx where xxx represents a string containing up to 70 printable characters with an ASCII code less than 128.

Request body:
xxxxxxxxxxx *(string corresponding to the value of "boundary" in the Content-Type)*
Content-Disposition: form-data; name=""; filename="*file_name.apk*"
Content-Type: application/vnd.android.package-archive
Binary content of the apk file
xxxxxxxxxxx *(string corresponding to the value of "boundary" in the Content-Type)*
Content-Disposition: form-data; name="lastModifiedDate"
Date in ISO 8601 format (e.g.: "2022-05-30T15:34:34.481Z")
xxxxxxxxxxx-- *(string corresponding to the value of "boundary" in the Content-Type, followed by two hyphens)*

Response status:

HTTP status	Description
201 Created	The request executed successfully. No content is available in the body of the response.
400 Bad request	At least one of the mandatory attributes of the request body is empty or missing.
	The JSON of the request body is syntactically incorrect.
	The API version provided as a parameter is incorrect.
401 Unauthorized	"Authorization" request header contains an incorrect or expired authentication token, or has no token.
403 Forbidden	The API does not support the context provided in the request.

404 Not Found	The URL is incorrectly formatted.
	The context provided as a parameter is incorrect.
405 Method Not Allowed	The request was sent using an incorrect method.
	The URL is incorrectly formatted.
409 Conflict	The application already exists in the catalog in the same version.
415 Unsupported Media Type	The "Content-Type" request header is missing or does not indicate "multipart/form-data".
	The file sent is not an APK file.

Response body:
None

23.20. API to add a file to the Android catalog

This API allows the calling application to add a file to the company's file catalog.

Method	URL
POST	**/public/api/v{version}/Catalogs/{typeOS}/Files**

Supported contexts:
Android,

Command line settings:

Settings	Description	Value
{version}	API version to use	1
{typeOS}	Context	android

Request headers:

Header	Content
Authorization	"Bearer" keyword followed by a space, and then the authentication token returned by the authentication API (without quotes)
Content-Type	multipart/form-data; boundary=xxxxxxxxxxx where xxx represents a string containing up to 70 printable characters with an ASCII code less than 128.

Request body:
xxxxxxxxxxx *(string corresponding to the value of "boundary" in the Content-Type)*
Content-Disposition: form-data; name=""; filename="*file_name.ext*"
Content-Type: *Mime type*
Binary content of the file
xxxxxxxxxxx *(string corresponding to the value of "boundary" in the Content-Type)*
Content-Disposition: form-data; name="lastModifiedDate"
Date in ISO 8601 format (e.g.: "2022-05-30T15:34:34.481Z")
xxxxxxxxxxx-- *(string corresponding to the value of "boundary" in the Content-Type, followed by two hyphens)*

Response status:

HTTP status	Description
201 Created	The request executed successfully. No content is available in the body of the response. If a file with the same name already exists in the catalog, it is replaced by the newly uploaded file.
400 Bad request	At least one of the mandatory attributes of the request body is empty or missing.
	The JSON of the request body is syntactically incorrect.
	The API version provided as a parameter is incorrect.
401 Unauthorized	"Authorization" request header contains an incorrect or expired authentication token, or has no token.
403 Forbidden	The API does not support the context provided in the request.
404 Not Found	The URL is incorrectly formatted.
	The context provided as a parameter is incorrect.
405 Method Not Allowed	The request was sent using an incorrect method.
	The URL is incorrectly formatted.
415 Unsupported Media Type	The "Content-Type" request header is missing or does not indicate "multipart/form-data".
	The type of the file sent is not authoried by Clyd for uploading.

Response body:
None

23.21. API to retrieve the list of kiosks

This API allows the calling application to retrieve the list of kiosks defined in the company.

Method	URL
GET	**/public/api/v{version}/Kiosks**

Supported contexts:
Dedicated Device or Device Admin company

Command line settings:

Settings	Description	Values
{version}	API version to use	1

Request headers:

Header	Content
Authorization	"Bearer" keyword followed by a space, and then the authentication token returned by the authentication API (without quotes)
Accept	application/json

Request body:
None

Response status:

HTTP status	Description

200 Success	The request executed successfully. Content is available in the body of the response.
400 Bad Request	The API version provided as a parameter is incorrect.
401 Unauthorized	"Authorization" request header contains an incorrect or expired authentication token, or has no token.
403 Forbidden	The API does not support the type of company in which it is authenticated.
404 Not Found	The URL is incorrectly formatted.
405 Method Not Allowed	The request was sent using an incorrect method.
	The URL is incorrectly formatted.

Response body (if HTTP 200 status):
Character string in JSON format containing a kiosk table:

```
[
 {
  "id": "integer",
  "name": "string",
  "deployedVersion": "string",
  "validationDateTime": "datetime",

  "lastDeploymentDateTime": "datetime"
 },
 ...
]
```

Internal ID of the kiosk
Kiosk name
Version deployed (0 if no version deployed)
Validation date of the kiosk in ISO 8601 format ("yyyy-mm-ddThh:mm:ss.n"), null if no date
Date of last deployment of the kiosk in ISO 8601 format ("yyyy-mm-ddThh:mm:ss.n"), null if no date

23.22. API to (partially) modify the properties of a kiosk

This API allows the calling application to modify the following properties of a kiosk:

- the list of authorized applications/activities (advanced properties)
- the list of prohibited applications/activities of the kiosk

Running this API saves changes to the kiosk draft. It does not cause it to be deployed.

- If a draft already exists when the API is called, the changes are saved in this draft (kiosk version is not incremented).
- If no draft exists, a new draft is created with a kiosk version incremented by 1.

Method	URL
PATCH	**/public/api/v{version}/Kiosks/{kioskId}**

Supported contexts:
Dedicated Device or Device Admin company

Command line settings:

Settings	Description	Value
{version}	API version to use	1
{kioskId}	Internal ID of the kiosk to be modified	

Request headers:

Header	Content

Authorization	"Bearer" keyword followed by a space, and then the authentication token returned by the authentication API (without quotes)
Content-Type	application/json

Request body:
String in JSON format containing the information to be modified in the kiosk.

```
{
  "authorizedApplications": [                    Table of authorized applications
  {
    "package": "string",                         Name of package to be authorized (required). Can be set to "*" to authorize all applications.
    "activity": "string",                        Activity to be authorized (required). Can be set to "*" to authorize all activities of the application.
    "comment": "string"                          Comment
  },
  ...
  ],
  "unauthorizedApplications": [                  Table of prohibited applications
  {
    "package": "string",                         Name of package to be prohibited (required)
    "activity": "string",                        Activity to be prohibited (required)
    "comment": "string"                          Comment
  },
  ...
  ]
}
```

Response status:

HTTP status	Description
204 No Content	The request executed successfully. No content is available in the body of the response.
400 Bad request	At least one of the mandatory attributes of the request body is empty or missing.
	The JSON of the request body is syntactically incorrect.
	The API version provided as a parameter is incorrect.
401 Unauthorized	"Authorization" request header contains an incorrect or expired authentication token, or has no token.
403 Forbidden	The API does not support the type of company in which it is authenticated.
404 Not Found	The URL is incorrectly formatted.
	The kiosk ID provided as a parameter is incorrect.
405 Method Not Allowed	The request was sent using an incorrect method.
	The URL is incorrectly formatted.
415 Unsupported Media Type	The "Content-Type" request header is missing or does not indicate "application/json".

Response body:
None

23.23. API to add an application/activity to the list of a kiosk's authorized applications

This API allows the calling application to add an application/activity pair to the list of the kiosk's authorized applications (advanced properties).

Running this API saves changes to the kiosk draft. It does not cause it to be deployed.

- If a draft already exists when the API is called, the changes are saved in this draft (kiosk version is not incremented).
- If no draft exists, a new draft is created with a kiosk version incremented by 1.

Method	URL
POST	**/public/api/v{version}/Kiosks/{kioskId}/AuthorizedApplications**

Supported contexts:
Dedicated Device or Device Admin company

Command line settings:

Settings	Description	Value
{version}	API version to use	1
{kioskId}	Internal ID of the kiosk to be modified	

Request headers:

Header	Content
Authorization	"Bearer" keyword followed by a space, and then the authentication token returned by the authentication API (without quotes)
Content-Type	application/json

Request body:
String in JSON format containing the information of the authorized application/activity to be added to the kiosk.

```
{
  "package": "string",
  "activity": "string",
  "comment": "string"
}
```

Package name (required) Can be set to "*" to authorize all applications.
Activity name (required). Can be set to "*" to authorize all activities of the application.
Comment

Response status:

HTTP status	Description
201 Created	The request executed successfully. No content is available in the body of the response.
400 Bad request	At least one of the mandatory attributes of the request body is empty or missing.
	The JSON of the request body is syntactically incorrect.
	The API version provided as a parameter is incorrect.
401 Unauthorized	"Authorization" request header contains an incorrect or expired authentication token, or has no token.
403 Forbidden	The API does not support the type of company in which it is authenticated.
404 Not Found	The URL is incorrectly formatted.
	The kiosk ID provided as a parameter is incorrect.

405 Method Not Allowed	The request was sent using an incorrect method.
	The URL is incorrectly formatted.
415 Unsupported Media Type	The "Content-Type" request header is missing or does not indicate "application/json".

Response body:
None

23.24. API to delete an application from the list of a kiosk's authorized applications

This API allows the calling application to delete all the application/activity pairs corresponding to the application passed as a parameter in the list of the kiosk's authorized applications (advanced properties).

Running this API saves changes to the kiosk draft. It does not cause it to be deployed.

- If a draft already exists when the API is called, the changes are saved in this draft (kiosk version is not incremented).
- If no draft exists, a new draft is created with a kiosk version incremented by 1.

Method	URL
DELETE	**/public/api/v{version}/Kiosks/{kioskId}/AuthorizedApplications?package={packageName}**

Supported contexts:
Dedicated Device or Device Admin company

Command line settings:

Settings	Description	Value
{version}	API version to use	1
{kioskId}	Internal ID of the kiosk to be modified	
{packageName}	Application package name	

Request headers:

Header	Content
Authorization	"Bearer" keyword followed by a space, and then the authentication token returned by the authentication API (without quotes)

Request body:
None

Response status:

HTTP status	Description
204 No Content	The request executed successfully. No content is available in the body of the response.
400 Bad request	The API version provided as a parameter is incorrect.
401 Unauthorized	"Authorization" request header contains an incorrect or expired authentication token, or has no token.

403 Forbidden	The API does not support the type of company in which it is authenticated.
404 Not Found	The URL is incorrectly formatted.
	The kiosk ID provided as a parameter is incorrect.
	The package name provided as a parameter does not match any application in the kiosk's list of authorized applications.
405 Method Not Allowed	The request was sent using an incorrect method.
	The URL is incorrectly formatted.

Response body:
None

23.25. API to add an application/activity to the list of a kiosk's prohibited applications

This API allows the calling application to add an application/activity pair to the kiosk's list of prohibited applications (advanced properties).

Running this API saves changes to the kiosk draft. It does not cause it to be deployed.

- If a draft already exists when the API is called, the changes are saved in this draft (kiosk version is not incremented).
- If no draft exists, a new draft is created with a kiosk version incremented by 1.

Method	URL
POST	**/public/api/v{version}/Kiosks/{kioskId}/UnauthorizedApplications**

Supported contexts:
Dedicated Device or Device Admin company

Command line settings:

Settings	Description	Value
{version}	API version to use	1
{kioskId}	Internal ID of the kiosk to be modified	

Request headers:

Header	Content
Authorization	"Bearer" keyword followed by a space, and then the authentication token returned by the authentication API (without quotes)
Content-Type	application/json

Request body:
String in JSON format containing the information of the prohibited application/activity to be added to the kiosk.

```
{
  "package": "string",
  "activity": "string",
  "comment": "string"
}
```

Package name (required)

Activity name (required). Can be set to "*" to authorize all activities of the application.

Comment

Response status:

HTTP status	Description
201 Created	The request executed successfully. No content is available in the body of the response.
400 Bad request	At least one of the mandatory attributes of the request body is empty or missing.
	The JSON of the request body is syntactically incorrect.
	The API version provided as a parameter is incorrect.
401 Unauthorized	"Authorization" request header contains an incorrect or expired authentication token, or has no token.
403 Forbidden	The API does not support the type of company in which it is authenticated.
404 Not Found	The URL is incorrectly formatted.
	The kiosk ID provided as a parameter is incorrect.
405 Method Not Allowed	The request was sent using an incorrect method.
	The URL is incorrectly formatted.
415 Unsupported Media Type	The "Content-Type" request header is missing or does not indicate "application/json".

Response body:
None

23.26. API to delete an application from the list of a kiosk's prohibited applications

This API allows the calling application to delete all the application/activity pairs corresponding to the application passed as a parameter in the list of prohibited applications of the kiosk (advanced properties).

Running this API saves changes to the kiosk draft. It does not cause it to be deployed.

- If a draft already exists when the API is called, the changes are saved in this draft (kiosk version is not incremented).
- If no draft exists, a new draft is created with a kiosk version incremented by 1.

Method	URL
DELETE	**/public/api/v{version}/Kiosks/{kioskId}/UnauthorizedApplications?package={packageName}**

Supported contexts:
Dedicated Device or Device Admin company

Command line settings:

Settings	Description	Value
{version}	API version to use	1
{kioskId}	Internal ID of the kiosk to be modified	
{packageName}	Application package name	

Request headers:

Header	Content
Authorization	"Bearer" keyword followed by a space, and then the authentication token returned by the authentication API (without quotes)

Request body:
None

Response status:

HTTP status	Description
204 No Content	The request executed successfully. No content is available in the body of the response.
400 Bad request	The API version provided as a parameter is incorrect.
401 Unauthorized	"Authorization" request header contains an incorrect or expired authentication token, or has no token.
403 Forbidden	The API does not support the type of company in which it is authenticated.
404 Not Found	The URL is incorrectly formatted.
	The kiosk ID provided as a parameter is incorrect.
	The package name provided as a parameter does not match any application in the kiosk's list of prohibited applications.
405 Method Not Allowed	The request was sent using an incorrect method.
	The URL is incorrectly formatted.

Response body:
None

23.27. API to deploy a kiosk

This API allows the calling application to request the deployment of a kiosk according to the deployment schedule defined in this kiosk (manual or scheduled deployment).

This API has the same behavior as clicking on the "Deploy v#" button in the kiosk details.

- If a draft of the kiosk exists when the API is called, this draft goes into production (version incremented) and the API launches deployment of this new version.
- If no draft of the kiosk exists, the API launches deployment of the current version.

Method	URL
POST	**/public/api/v{version}/Kiosks/{kioskId}/Deploy**

Supported contexts:
Dedicated Device or Device Admin company

Command line settings:

Settings	Description	Value
{version}	API version to use	1
{kioskId}	Internal ID of the kiosk to be deployed	

Request headers:

Header	Content
Authorization	"Bearer" keyword followed by a space, and then the authentication token returned by the authentication API (without quotes)

Request body:
None

Response status:

HTTP status	Description
204 No Content	The request executed successfully. No content is available in the body of the response. This status ensures that the kiosk deployment has been requested but does not ensure it was executed successfully.
400 Bad request	The API version provided as a parameter is incorrect.
401 Unauthorized	"Authorization" request header contains an incorrect or expired authentication token, or has no token.
403 Forbidden	The API does not support the type of company in which it is authenticated.
404 Not Found	The URL is incorrectly formatted.
	The kiosk ID provided as a parameter is incorrect.
405 Method Not Allowed	The request was sent using an incorrect method.
	The URL is incorrectly formatted.

Response body:
None

23.28. API to retrieve the list of WorkSpaces

This API allows the calling application to retrieve the list of WorkSpaces defined in the company.

Method	URL
GET	**/public/api/v{version}/WorkSpaces**

Supported contexts:
Dedicated Device or Device Admin company

Command line settings:

Settings	Description	Values
{version}	API version to use	1

Request headers:

Header	Content
Authorization	"Bearer" keyword followed by a space, and then the authentication token returned by the authentication API (without quotes)
Accept	application/json

Request body:
None

Response status:

HTTP status	Description
200 Success	The request executed successfully. Content is available in the body of the response.
400 Bad Request	The API version provided as a parameter is incorrect.
401 Unauthorized	"Authorization" request header contains an incorrect or expired authentication token, or has no token.
403 Forbidden	The API does not support the type of company in which it is authenticated.
404 Not Found	The URL is incorrectly formatted.
405 Method Not Allowed	The request was sent using an incorrect method.
	The URL is incorrectly formatted.

Response body (if HTTP 200 status):
Character string in JSON format containing a WorkSpace table:

```
[
 {
  "id": "integer",                          Internal ID of the WorkSpace
  "name": "string",                         WorkSpace name
  "deployedVersion": "string",              Version deployed (0 if no version deployed)
  "validationDateTime": "datetime",         Validation date of the WorkSpace in ISO 8601 format
                                            ("yyyy-mm-ddThh:mm:ss.n"), null if no date
  "lastDeploymentDateTime": "datetime"      Date of last deployment of the WorkSpace in
 },                                         ISO 8601 format ("yyyy-mm-ddThh:mm:ss.n"), null if
 ...                                        no date
]
```

23.29. API to add a shortcut to a WorkSpace page

This API allows the calling application to add a shortcut (application, widget, link to a website, command) to a page in a WorkSpace. This shortcut will be added after the shortcuts already present on the page.

Running this API saves changes to the WorkSpace draft. It does not cause it to be deployed.

- If a draft already exists when the API is called, the changes are saved in this draft (WorkSpace version is not incremented).
- If no draft exists, a new draft is created with a WorkSpace version incremented by 1.

Method	URL
POST	**/public/api/v{version}/WorkSpaces/{workSpaceId}/Pages/{pageIndex}/Shortcuts/**

Supported contexts:
Dedicated Device or Device Admin company

Command line settings:

Settings	Description	Value
{version}	API version to use	1
{workSpaceId}	Internal ID of the WorkSpace to be modified	
{pageIndex}	Index of the WorkSpace page where the shortcut should be added	1 to n *(index 1 for the first page)*

Request headers:

Header	Content
Authorization	"Bearer" keyword followed by a space, and then the authentication token returned by the authentication API (without quotes)
Content-Type	application/json

Request body:

String in JSON format containing the information of the shortcut to be added to the WorkSpace page:

```
{
  "type": "string",
  "link": "string",

  "label": "string",
  "parameter": "string"
..”position”: “integer”
}
```

Type of shortcut (required)
Shortcut identification (required, value depends on the type of shortcut)
Label associated with the shortcut
Additional parameter (depending on the type of shortcut)
Optional parameter to define an icon position between 1 and 512 (requires Workspace with automatic position disabled)

For a "Play Store application" shortcut:

- (Required) The "type" attribute is set to "**play-store-app**".
- (Required) The "link" attribute contains the application's package name.
- The "label" attribute lets you customize the shortcut's label. Default: name of application.
- The "parameter" attribute is not used.

For an "In-house application" shortcut:

- (Required) The "type" attribute is set to "**in-house-app**".
- (Required) The "link" attribute contains the application's package name.
- The "label" attribute lets you customize the shortcut's label. Default: name of the application.
- (Required) The "parameter" attribute is required if several versions of the application exist in the catalog of in-house applications. In this case, it must contain the version code of the package to be added as a shortcut.

For an "Inventoried application" shortcut:

- (Required) The "type" attribute is set to "**inventoried-app**".
- (Required) The "link" attribute contains the application's package name.
- The "label" attribute lets you customize the shortcut's label. Default: name of the application.
- (Required) The "parameter" attribute is required if several versions of the application exist across the company's device software inventories. In this case, it must contain the version code of the package to be added as a shortcut.

For a "File" shortcut:

- (Required) The "type" attribute is set to "**file**".
- (Required) The "link" attribute contains the filename of the file catalog.
- The "label" attribute is not used.
- The "parameter" attribute is not used.

For a "Widget" shortcut:

- (Required) The "type" attribute is set to "**widget**".
- (Required) The "link" attribute is used to indicate the type of widget and contains one of the following values:
    - "wifi": Wi-Fi widget
    - "3g": mobile data widget
    - "ringtone": ringtone widget
    - "phone": phone widget
    - "bluetooth": Bluetooth widget
    - "location": location widget
    - "rotation": automatic rotation widget
    - "connect": server connection widget

    - "message": message widget
    - "wifi-config" : widget for Wi-Fi configuration
- The "label" attribute lets you customize the shortcut's label. Default: English name of the widget.
- The "parameter" attribute is only used for the message widget to indicate the content of the message to be displayed on the device.

For a "Link to a website" shortcut:

- (Required) The "type" attribute is set to "**website**".
- (Required) The "link" attribute contains the URL of the website and must begin with "http://" or "https://".
- (Required) The "label" attribute contains the label of the shortcut.
- The "parameter" attribute is not used.

For a "Command" shortcut:

- (Required) The "type" attribute is set to "**command**".
- (Required) The "link" attribute contains the command to be run.
- (Required) The "label" attribute contains the label of the shortcut.
- The "parameter" attribute is not used.

Response status:

HTTP status	Description
201 Created	The request executed successfully. No content is available in the body of the response.
400 Bad request	At least one of the mandatory attributes of the request body is empty or missing.
	The JSON of the request body is syntactically incorrect.
	The API version provided as a parameter is incorrect.
	For a "Link to a website" shortcut, the URL provided in the "link" attribute of the request body does not begin with "http://" or "https://".
	For an "In-house application" shortcut, there are several versions of this application in the in-house catalog but the version code of the application has not been provided in the "parameter" attribute of the request body.
	For an "Inventoried application" shortcut, there are several versions of this application across the company's device software inventories, but the version code of the application has not been provided in the "parameter" attribute of the request body.
401 Unauthorized	"Authorization" request header contains an incorrect or expired authentication token, or has no token.
403 Forbidden	The API does not support the type of company in which it is authenticated.
	A Play Store application cannot be added to the WorkSpace while authenticated in a Device Admin company.
404 Not Found	The URL is incorrectly formatted.
	The WorkSpace ID provided as a parameter is incorrect.
	The page index provided as a parameter is incorrect (out of range).
	For a "Play Store application" shortcut, the application whose package name is provided in the "link" attribute of the request body does not exist in the company's enterprise store.
	For an "In-house application" shortcut, the application whose package name is provided in the "link" attribute, and whose version code may be provided in the "parameter" attribute of the request body, does not exist in the company's in-house catalog.
	For an "Inventoried application" shortcut, the application whose package name is provided in the "link" attribute, and whose version code may be provided in the

	"parameter" attribute of the request body, does not exist in the company's in-house catalog.
	The file whose name is provided in the request body does not exist in the CLYD file catalog.
405 Method Not Allowed	The request was sent using an incorrect method.
	The URL is incorrectly formatted.
409 Conflict	The application you are trying to add is already present in the WorkSpace in a different version.
415 Unsupported Media Type	The "Content-Type" request header is missing or does not indicate "application/json".

Response body:
None

23.30. API to delete a shortcut from a WorkSpace page

This API allows the calling application to delete a shortcut (application, widget, link to a website, command) from a page in a WorkSpace.

Running this API saves changes to the WorkSpace draft. It does not cause it to be deployed.

- If a draft already exists when the API is called, the changes are saved in this draft (WorkSpace version is not incremented).
- If no draft exists, a new draft is created with a WorkSpace version incremented by 1.

Method	URL
DELETE	**/public/api/v{version}/WorkSpaces/{workSpaceId}/Pages/{pageIndex}/Shortcuts ?type={shortcutType}&{otherParameters}**

Supported contexts:
Dedicated Device or Device Admin company

Command line settings:

Settings	Description	Value
{version}	API version to use	1
{workSpaceId}	Internal ID of the WorkSpace to be modified	
{pageIndex}	Index of the WorkSpace page from which the shortcut has to be removed	1 to n *(index 1 for the first page)*
{type}	Type of shortcut	app file widget website command
{otherParameters}	Other parameters, depending on the type of widget: • "app": ○ **link={packageName}** to delete all application shortcuts with the package name {packageName} ○ **link={packageName}&label={label}** to delete all application shortcuts with the	{widgetType} can be set to the following values: wifi 3g ringtone phone

	package name {packageName} and label {label}
	• "file": **link={fileName}** to delete all file shortcuts with the name {fileName}
	• "widget":
	  ○ **link={widgetType}** to delete all widget shortcuts of the type {widgetType}
	  ○ **link={widgetType}&label={label}** to delete all widget shortcuts of the type {widgetType} and with the label {label}
	• "website": **label={label}** to delete all links to websites with the label {label}
	• "command": **label={label}** to delete all commands with the label {label}
		bluetooth
		location
		rotation
		connect
		message
		wifi-config

Request headers:

Header	Content
Authorization	"Bearer" keyword followed by a space, and then the authentication token returned by the authentication API (without quotes)

Request body:
None

Response status:

HTTP status	Description
204 No Content	The request executed successfully. No content is available in the body of the response.
400 Bad request	At least one of the mandatory attributes of the request body is empty or missing.
	The JSON of the request body is syntactically incorrect.
	The API version provided as a parameter is incorrect.
	The type of shortcut provided as a parameter has an unrecognised value.
	For a "Widget" type shortcut, the widget type provided as a parameter has an unrecognised value.
401 Unauthorized	"Authorization" request header contains an incorrect or expired authentication token, or has no token.
403 Forbidden	The API does not support the type of company in which it is authenticated.
404 Not Found	The URL is incorrectly formatted.
	The WorkSpace ID provided as a parameter is incorrect.
	The page index provided as a parameter is incorrect (out of range).
	No shortcuts match the criteria specified in the request parameters.
405 Method Not Allowed	The request was sent using an incorrect method.
	The URL is incorrectly formatted.

Response body:
None

23.31. API to deploy a WorkSpace

This API allows the calling application to request the deployment of a WorkSpace according to the deployment schedule defined in this WorkSpace (manual or scheduled deployment).

This API has the same behavior as clicking on the "Deploy v#" button in the WorkSpace details.

- If a draft of the WorkSpace exists at the time of the API call, this draft goes into production (version incremented) and the API launches deployment of this new version.
- If no draft of the WorkSpace exists, the API launches deployment of the current version.

Method	URL
POST	**/public/api/v{version}/WorkSpaces/{workSpaceId}/Deploy**

Supported contexts:
Dedicated Device or Device Admin company

Command line settings:

Settings	Description	Value
{version}	API version to use	1
{workSpaceId}	Internal ID of WorkSpace to be deployed	

Request headers:

Header	Content
Authorization	"Bearer" keyword followed by a space, and then the authentication token returned by the authentication API (without quotes)

Request body:
None

Response status:

HTTP status	Description
204 No Content	The request executed successfully. No content is available in the body of the response. This status ensures that the WorkSpace deployment has been requested but does not ensure that it has been successfully run.
400 Bad request	The API version provided as a parameter is incorrect.
401 Unauthorized	"Authorization" request header contains an incorrect or expired authentication token, or has no token.
403 Forbidden	The API does not support the type of company in which it is authenticated.
404 Not Found	The URL is incorrectly formatted.
	The WorkSpace ID provided as a parameter is incorrect.
405 Method Not Allowed	The request was sent using an incorrect method.
	The URL is incorrectly formatted.

Response body:
None

23.32. API to retrieve company license information

This API allows the calling application to retrieve the company's Clyd/MediaCOntact license information.

Method	URL
GET	**/public/api/v{version}/LicenseInformation**

Command line parameters:

Parameter	Description	Value
{version}	API version to use	1

Request headers:

Header	Content
Authorization	Keyword "Bearer" followed by a space then the authentication token returned by the authentication API (without quotes)
Accept	application/json

Request body :
None

Response status:

HTTP Status	Description
200 Success	The query ran successfully. Content is available in the response body.
400 Bad Request	The API version provided as a parameter is incorrect.
401 Unauthorized	The "Authorization" request header contains a bad or expired authentication token, or does not contain a token.
404 Not Found	The URL is incorrectly formatted.
	The type of devices provided in parameter is incorrect.
	A global root certificate was entered on the server, but no company certificate was entered for the company.
405 Method Not Allowed	The request was sent using an incorrect method.
	The URL is incorrectly formatted.

Response body (if http status 200):
Character string in JSON format containing the license information:

```
{
  "licenseType" : "string",
  "evaluationMode" : bool,
  "serialNumber" : "string",
  "expirationDate" : "datetime",

  "devices" : [
    {
      "osType" : "string",

      "authorized" : integer,
      "assigned" : integer
    },
    ...
  ]
}
```

License type ("global", "company")
Evaluation mode (true, false)
Serial number
License expiration date/time in ISO 8601 format ("aaaa-mm-qqThh:mi:ss.n")
Licenses by device type

Device type ("all", "android", "windows", "printer")
Number of devices authorized by the license
Number of assigned devices

Example of a server with registered global license without distribution by type of devices:

```
{
  "licenseType" : "global",
  "evaluationMode" : false,
```

```
  "serialNumber" : "abcdabcdabcdabcd,
  "expirationDate" : "2023-12-01T00:00:00.000",
  "devices" : [
    {
      "osType" : "all",
      "authorized" : 150,
      "assigned" : 79
    }
  ]
}
```


Example of a server with registered company license with distribution by type of devices:


```
{
  "licenseType" : "company",
  "evaluationMode" : false,
  "serialNumber" : "abcdabcdabcdabcd,
  "expirationDate" : "2023-12-01T00:00:00.000",
  "devices" : [
    {
      "osType" : "android",
      "authorized" : 1500,
      "assigned" : 1235
    },
    {
      "osType" : "printers",
      "authorized" : 200,
      "assigned" : 120
    },
    {
      "osType" : "windows",
      "authorized" : 0,
      "assigned" : 0
    },
  ]
}
```


Example of a server in evaluation mode:


```
{
  "licenseType" : "global",
  "evaluationMode" : true,
  "serialNumber" : null,
  "expirationDate" : date,
  "devices" : [
    {
      "osType" : "all",
      "authorized" : null,
      "assigned" : integer
    }
  ]
}
```

END OF DOCUMENT